Australian Government
Department of Defence
Defence Science and
Technology Organisation

Minimum Paths to Interception of a Moving Target when Constrained by Turning Radius

Jason R. Looker

Air Operations Division

Defence Science and Technology Organisation

DSTO–TR–2227

ABSTRACT

Entities in some simulations of military operations move unrealistically from point to point and are not constrained by their turning radius. The fidelity of this representation may be insufficient for operations research studies. In this paper a pursuer intercepting a target is considered, where the pursuer and target are moving at constant speeds in two dimensions and the target has a constant velocity. The minimum feasible path to interception for a given turning radius is sought. A rigorous analysis of the model constraints produced an algorithm that can be used to systematically search the feasible region for the minimum path to interception. At the core of the algorithm is a single implicit equation for the minimum time to interception. This enables the effect of turning radius to be incorporated as a constraint into simulations of military operations, improving their fidelity. The algorithm is also straightforward to implement when compared with, for example, a traditional flight dynamics model, and has a broad range of applications in path optimisation problems, the development of computer games and robotics.

Published by

DSTO Defence Science and Technology Organisation
506 Lorimer St,
Fishermans Bend, Victoria 3207, Australia

Telephone: (03) 9626 7000
Facsimile: (03) 9626 7999

AR No. AR 014–359
December, 2008

APPROVED FOR PUBLIC RELEASE

Minimum Paths to Interception of a Moving Target when Constrained by Turning Radius

Executive Summary

Entities in some simulations of military operations move unrealistically from point to point and are not constrained by their turning radius. The fidelity of this representation may be insufficient for operations research studies.

In this paper a pursuer intercepting a target is considered, where the pursuer and target are moving at constant speeds in two dimensions and the target has a constant velocity. The minimum feasible path to interception for a given turning radius is sought. The pursuer must obey three constraints:

(C1) The pursuer cannot reverse its direction to intercept a target.

(C2) The pursuer cannot intercept a target inside its turning-circle.

(C3) The pursuer may perform at most one complete turn.

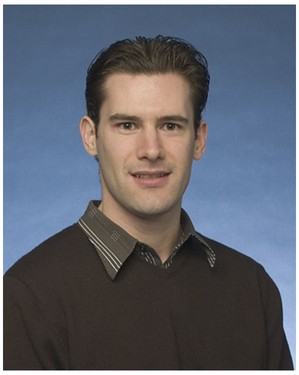
It is assumed that the pursuer's speed is strictly greater than the target's speed. This assumption is not absolutely necessary, however, it substantially simplifies the analysis and discussion.

In the present work, a rigorous analysis of Constraints (C1)–(C3) produced an algorithm that can be used to systematically search the feasible region for the minimum path to interception. At the core of the algorithm is a single implicit equation for the minimum time to interception. This equation is valid in an arbitrary Cartesian coordinate system and encompasses both left and right turns. For the purpose of validation, the algorithm has been implemented as a Mathematica package that displays the minimum feasible path to interception.

Three methods are proposed for incorporating classification range into the present model: an exact method, a heuristic method, and a method that includes an angle of approach. These methods provide simple models of a pursuer's sensor performance. The present model can be easily modified to encompass the heuristic and angle of approach methods.

The point-to-point and unconstrained movement of entities in some simulations of military operations is unrealistic. The algorithm developed here enables the effect of turning radius to be incorporated as a constraint into these simulations, improving their fidelity. The algorithm is also straightforward to implement when compared with, for example, a traditional flight dynamics model, and has a broad range of applications in path optimisation problems, the development of computer games and robotics.

Author



Jason Looker
Air Operations Division

Dr Jason Looker joined DSTO in 2006 as an Operations Research Scientist after completing a BSc (Honours) and PhD in Mathematics from The University of Melbourne. He is a member of the AOD team supporting RAAF Experimentation, and is leading a research project on the path planning of low-observable aircraft through hostile environments.

Contents

Appendices

Figures

Tables

Notation

$\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$	standard basis vectors in the x, y and z directions
$\mathbf{x}$	a generic 3-d vector, $\mathbf{x} = (x_1, x_2, x_3)$
$\lvert\mathbf{x}\rvert$	magnitude of $\mathbf{x}$, $\lvert\mathbf{x}\rvert = \sqrt{x_1^2 + x_2^2 + x_3^2}$
$\mathbf{x} \cdot \mathbf{y}$	scalar (dot) product, $\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + x_3 y_3$
$\mathbf{x} \times \mathbf{y}$	vector (cross) product, $\mathbf{x} \times \mathbf{y} = (x_2 y_3 - x_3 y_2, x_3 y_1 - x_1 y_3, x_1 y_2 - x_2 y_1)$
n	$n = 0$ results in a left turn, $n = 1$ results in a right turn
$\mathbf{x}_c$	centre of the pursuer's turning-circle
r_c	radius of the pursuer's turning-circle
$\bar{r}_{cl}$	pursuer's (dimensionless) classification range
$\mathbf{x}_{\mathrm{in}}$	initial position of the pursuer
$\mathbf{u}_{\mathbf{P}}^0$	initial velocity of the pursuer
$C_{\mathbf{P}}$	speed of the pursuer
$\mathbf{x}_{\mathrm{out}}$	position where the pursuer exits its turn
$\mathbf{x}_{\mathbf{T}}^0$	initial position of the target
$\mathbf{x}_{\mathbf{T}}(t)$	position of the target at time t
$\mathbf{u}_{\mathbf{T}}$	velocity of the target
$C_{\mathbf{T}}$	speed of the target
$\mathbf{x}_I$	point of interception
$\mathbf{x}_{cl}$	point of classification
α	$\left\lvert\mathbf{x}_{\mathbf{T}}^0 - \mathbf{x}_c\right\rvert^2 - 1$ (in dimensionless form)
β	$\mathbf{u}_{\mathbf{T}} \cdot \left(\mathbf{x}_{\mathbf{T}}^0 - \mathbf{x}_c\right)$ (in dimensionless form)
ϵ	$C_{\mathbf{T}}/C_{\mathbf{P}}$
t_c	time for the pursuer to complete its turn
t_l	time for the pursuer to move from $\mathbf{x}_{\mathrm{out}}$ to $\mathbf{x}_I$
T	total time to interception, $t_c + t_l$
T_{cl}	total time to classification
T_L	time for the target to enter the pursuer's turning-circle [see Equation (21)]
T_R	time for the target to exit the pursuer's turning-circle [see Equation (21)]
T_0	time to interception if the pursuer does not turn [see Equation (22)]
$\overline{T}_0$	time to interception if the pursuer does not turn [see Equation (A5)]
T_π	time to interception if the pursuer takes π units of time to turn [see Equation (23)]
$\overline{T}_\pi$	time to interception if the pursuer takes π units of time to turn [see Equation (24)]
θ_c	angle between $\mathbf{x}_{\mathrm{in}} - \mathbf{x}_c$ and $\mathbf{x}_{\mathrm{out}} - \mathbf{x}_c$

1 Introduction

Entities in some simulations of military operations move unrealistically from point to point and are not constrained by their turning radius. The fidelity of this representation may be insufficient for operations research studies.

In this paper a pursuer intercepting a target is considered, where the pursuer and target are moving at constant speeds in two dimensions and the target has a constant velocity. The minimum feasible path to interception for a given turning radius is sought. The aim is to rigorously develop an algorithm that enables the effect of turning radius to be incorporated as a constraint into simulations of military operations, improving their fidelity. This algorithm should be straightforward to implement when compared with, for example, a traditional flight dynamics model, and have a broad range of applications in path optimisation problems, the development of computer games and robotics.

During maritime surveillance operations, aircraft search areas of interest in order to classify as many ships as possible in the shortest possible time. Marlow, Kilby & Mercer [2007] endeavour to optimise maritime surveillance operations by comparing the effect of various search algorithms on the surveillance aircraft's performance. At present their model's aircraft simply flies from point to point and is not constrained by its turning radius.

The impact of turning radius and classification range on the optimal route length in a simplified maritime surveillance scenario has been modelled by Mercer et al. [2008], using a method based on trigonometry. A similar method is also used by computer game developers [Pinter 2001, Pinter 2002]. Both of these methods require solving a system of nonlinear equations, where the feasible region is not determined and the resulting curves are not necessarily minimum feasible paths to interception.

Classical pursuit curves are obtained when a point A moves with constant speed towards another point B moving with constant speed along a known curve, where the tangent vector at A is required to be parallel to the line connecting A and B [Boole 1859, Colman 1991, Eliezer & Barton 1992, Eliezer & Barton 1995, Barton & Eliezer 2000, Weisstein 2008]. In this case, the turning radius of A is not constrained and the resulting curves are also not necessarily minimum feasible paths to interception. In the present work, the tangent vector of the pursuer is not required to be parallel to the line connecting the pursuer to the target.

The objective of pursuit-evasion games is to determine the optimal strategies that result in a pursuer capturing an evader. The theory of pursuit-evasion games was first studied by Isaacs [1965] and has been applied to a variety of problems in the guidance and optimal control literature: for example, see Shinar, Guelman & Green [1989] and Shima & Shinar [2002]. The present work was motivated by research into maritime surveillance operations [Marlow, Kilby & Mercer 2007]. In this instance, the speed of the surveillance aircraft is typically much greater than the speed of the ship, and consequently any evasive manoeuvres executed by the ship will have little impact on the surveillance aircraft's ability to classify the ship. For this reason, the target considered here does not attempt to evade the pursuer, and although this can be regarded as a special case of a pursuit-evasion game, a simpler, more direct method will be used to construct a solution.

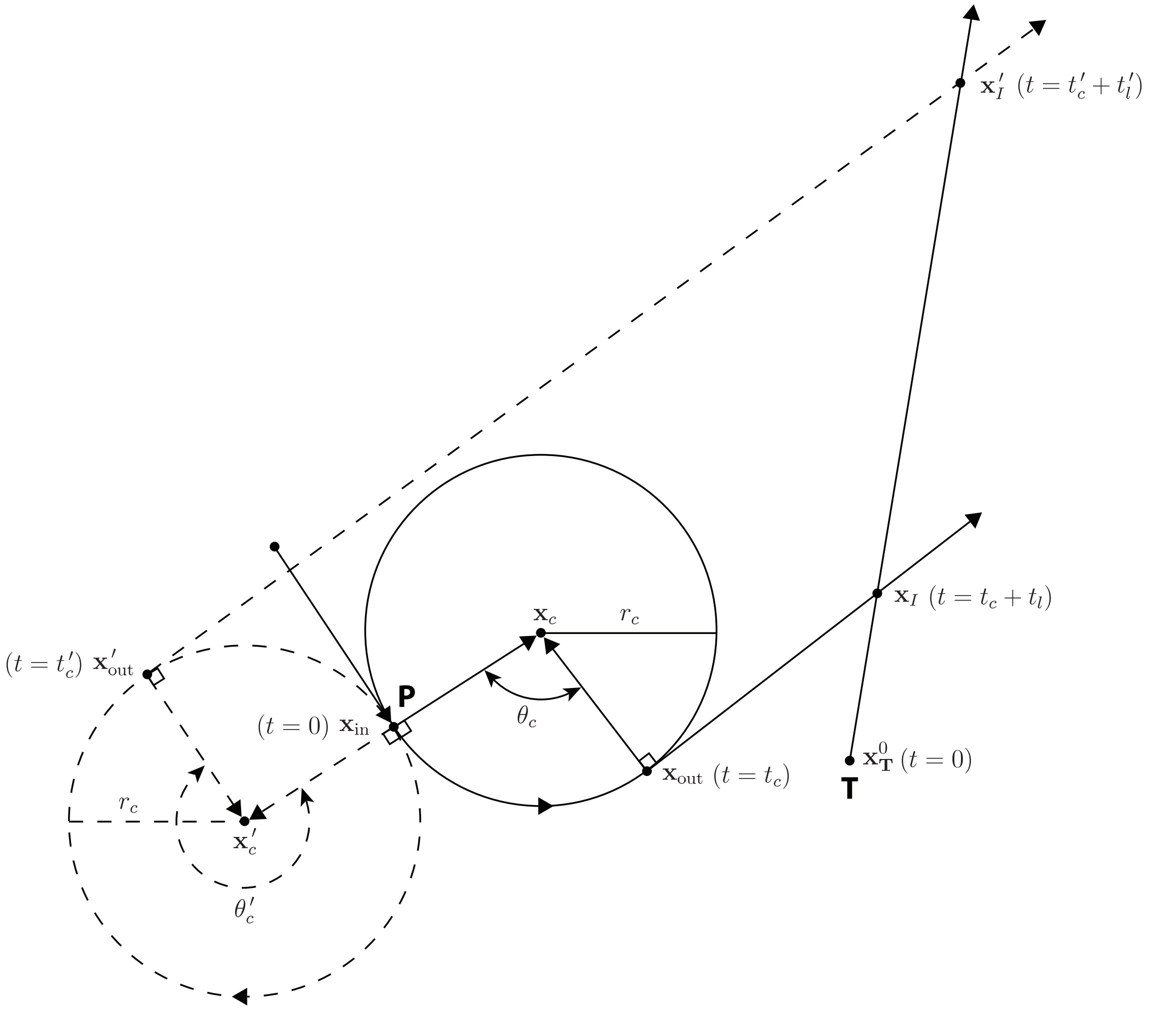

$\mathbf{x}_I'$ $(t = t_c' + t_l')$
$\mathbf{x}_c$
r_c
$\mathbf{x}_I$ $(t = t_c + t_l)$
$(t = t_c')$ $\mathbf{x}_{\text{out}}'$
P
$(t = 0)$ $\mathbf{x}_{\text{in}}$
θ_c
$\mathbf{x}_{\text{out}}$ $(t = t_c)$
$\mathbf{x}_{\mathbf{T}}^0$ $(t = 0)$
T
r_c
$\mathbf{x}_c'$
θ_c'

Figure 1: *Definition diagram showing two possible paths to interception resulting from left and right turns. The pursuer (P) turns to intercept the target (T) at time* $t = 0$*. The pursuer's initial velocity is* $\mathbf{u}_{\mathbf{P}}^0$*, and the target's velocity is* $\mathbf{u}_{\mathbf{T}}$*. The pursuer and target have constant speeds* $C_{\mathbf{P}}$ *and* $C_{\mathbf{T}}$*, respectively.*

Dubins [1957] studied planar continuously differentiable curves of minimal length with average curvature bounded by R^{-1}, between prescribed initial and final positions and orientations. He proved that such curves exist and are necessarily a subpath of a path of type CLC or of type CCC, where C is an arc of a circle of radius R, and L is a straight line segment. These results can be applied to the motion planning of a car-like robot that only moves forwards. Reeds & Shepp [1990] have extended the work of Dubins [1957] to allow for both forwards and backwards motion (that is, paths with cusps), and provide explicit formulae for the resulting 68 paths. There are many other generalisations of Dubins' paper: for example, to the motion planning of a car-like robot moving amid obstacles [Laumond et al. 1994], and to the Traveling Salesperson Problem for a car-like robot [Savla, Frazzoli & Bullo 2008]. The paths considered in this paper differ from Dubins' paths, as the pursuer's final position and orientation are not prescribed.

Not all paths to interception will be considered in this paper. Instead, as the final orientation of the pursuer is not prescribed and since the shortest path between two points is a straight line, the work of Dubins [1957] suggests that minimal paths to interception will consist of an arc of a circle of radius equal to the turning radius of the pursuer, followed by a straight line segment.

The definition diagram for the present scenario is shown in Figure 1. At time $t = 0$ the pursuer at position $\mathbf{x}_{\text{in}}$ with velocity $\mathbf{u}_{\mathbf{P}}^{0}$ turns with a turning radius of r_c to intercept the target, which is initially at position $\mathbf{x}_{\mathbf{T}}^{0}$ with a constant velocity $\mathbf{u}_{\mathbf{T}}$. It may be possible for the pursuer to turn either left or right to intercept the target. The pursuer exits its turn at position $\mathbf{x}_{\text{out}}$ $(t = t_c)$ and intercepts the target at position $\mathbf{x}_I$ $(t = t_c + t_l)$. The pursuer and target move in two dimensions and have constant speeds $C_{\mathbf{P}}$ and $C_{\mathbf{T}}$, respectively. The minimum feasible time to interception, and hence the minimum feasible path to interception, for a given turning radius is sought. The pursuer must obey three constraints:

(C1) The pursuer cannot reverse its direction to intercept a target.

(C2) The pursuer cannot intercept a target inside its turning-circle.

(C3) The pursuer may perform at most one complete turn.

It is assumed that the pursuer's speed is strictly greater than the target's speed. This assumption is not absolutely necessary, however, it substantially simplifies the analysis and discussion.

In maritime surveillance operations, aircrew can classify a ship if the aircraft is within a certain distance of the ship, which depends on the aircraft's sensor suite, weather, sea state, *et cetera*. Three methods will be proposed for incorporating classification range into the present scenario: an exact method, a heuristic method, and a method that includes an angle of approach. These methods provide simple models of a pursuer's sensor performance.

2 The mathematical model

In this section, a mathematical model of the scenario described in the Introduction is derived.

2.1 Derivation of the governing equations

Any point $\mathbf{x}$ on a circle of radius r_c centred on $\mathbf{x}_c$ satisfies

$$|\mathbf{x} - \mathbf{x}_c| = r_c.$$

Since $\mathbf{x}_c - \mathbf{x}_{\text{in}}$ is orthogonal to $\mathbf{u}^0_{\mathbf{P}}$ (see Figure 1) it follows that $\mathbf{x}_c$ satisfies

$$|\mathbf{x}_{\text{in}} - \mathbf{x}_c| = r_c,$$

$$(\mathbf{x}_c - \mathbf{x}_{\text{in}}) \cdot \mathbf{u}^0_{\mathbf{P}} = 0.$$

Using the aforementioned reasoning, the exit point on the turning-circle obeys

$$|\mathbf{x}_{\text{out}} - \mathbf{x}_c| = r_c,$$

$$(\mathbf{x}_c - \mathbf{x}_{\text{out}}) \cdot (\mathbf{x}_I - \mathbf{x}_{\text{out}}) = 0,$$

where the point of interception is given by

$$\mathbf{x}_I = \mathbf{x}^0_{\mathbf{T}} + T\mathbf{u}_{\mathbf{T}}.$$

Here $T = t_c + t_l$ is the total time to interception.

Since the pursuer is moving at a constant speed, the time taken for the pursuer to complete the turn is

$$t_c = \frac{\theta_c r_c}{C_{\mathbf{P}}},$$

where θ_c is the angle between $\mathbf{x}_{\text{in}} - \mathbf{x}_c$ and $\mathbf{x}_{\text{out}} - \mathbf{x}_c$ (see Figure 1). The definition of the scalar product yields an expression for θ_c [Spiegel 1974]:

$$\cos(\theta_c) = \frac{(\mathbf{x}_{\text{in}} - \mathbf{x}_c) \cdot (\mathbf{x}_{\text{out}} - \mathbf{x}_c)}{r_c^2}.$$

Once again since the pursuer is moving at a constant speed, the time taken for the pursuer to move from the exit point to the point of interception is

$$t_l = \frac{|\mathbf{x}_I - \mathbf{x}_{\text{out}}|}{C_{\mathbf{P}}}.$$

For convenience and to highlight the key parameters of the problem, we introduce the following dimensionless variables:

$$\mathbf{x} = r_c\hat{\mathbf{x}}, \qquad t = \frac{r_c}{C_{\mathbf{P}}}\hat{t}, \qquad \mathbf{u}^0_{\mathbf{P}} = C_{\mathbf{P}}\hat{\mathbf{u}}^0_{\mathbf{P}}, \qquad \mathbf{u}_{\mathbf{T}} = C_{\mathbf{T}}\hat{\mathbf{u}}_{\mathbf{T}}, \tag{1}$$

where a caret indicates a dimensionless variable and $\mathbf{x}$ is a generic position vector. Employing these scales results in the following dimensionless system:

$$|\mathbf{x}_{\text{in}} - \mathbf{x}_c| = 1, \tag{2}$$

$$(\mathbf{x}_c - \mathbf{x}_{\text{in}}) \cdot \mathbf{u}^0_{\mathbf{P}} = 0, \tag{3}$$

$$|\mathbf{x}_{\text{out}} - \mathbf{x}_c| = 1, \tag{4}$$

$$(\mathbf{x}_c - \mathbf{x}_{\text{out}}) \cdot (\mathbf{x}_I - \mathbf{x}_{\text{out}}) = 0, \tag{5}$$

$$\mathbf{x}_I = \mathbf{x}^0_{\mathbf{T}} + \epsilon T \mathbf{u}_{\mathbf{T}}, \tag{6}$$

$$T = t_c + t_l, \tag{7}$$

$$\cos(t_c) = (\mathbf{x}_{\text{in}} - \mathbf{x}_c) \cdot (\mathbf{x}_{\text{out}} - \mathbf{x}_c), \tag{8}$$

$$t_l = |\mathbf{x}_I - \mathbf{x}_{\text{out}}|\,, \tag{9}$$

where the dimensionless parameter ϵ is defined by[1]

$$\epsilon = \frac{C_{\mathbf{T}}}{C_{\mathbf{P}}}.$$

Note that the carets have been omitted for convenience, thus *all functions, variables and parameters will henceforth refer to dimensionless quantities.*

2.2 Solution of the governing equations

Vectors of the form $\mathbf{X} = (a, b, 0)$ have the following properties:

1. $|\mathbf{X} \times \mathbf{k}| = |\mathbf{X}|$;
2. $(\mathbf{X} \times \mathbf{k}) \cdot \mathbf{X} = 0$;
3. $(\mathbf{X} \times \mathbf{k}) \times \mathbf{k} = -\mathbf{X}$;
4. $(\mathbf{X} \times \mathbf{k}) \times \mathbf{X} = |\mathbf{X}|^2\,\mathbf{k}$,

where $\mathbf{k} = (0, 0, 1)$. Properties 1 and 2 can be verified by inspection. Properties 3 and 4 can be derived using the following identity for general 3-dimensional vectors [Spiegel 1974]:

$$(\mathbf{A} \times \mathbf{B}) \times \mathbf{C} = (\mathbf{A} \cdot \mathbf{C})\,\mathbf{B} - (\mathbf{B} \cdot \mathbf{C})\,\mathbf{A}.$$

Properties 1 to 4 will now be used to solve Equations (2) to (9).

[1] A stationary target implies that $\epsilon = 0$. In this case Equations (2) to (9) decouple and an explicit solution for the total time to interception can be obtained. An approximate expression has been derived that is valid in the limit $\epsilon \rightarrow 0$. However, this expression is cumbersome and only accurate for very small values of ϵ.

2.2.1 The exit point on the turning-circle

It can be verified using Properties 1 and 2 that the solution to Equations (2) and (3) is

$$\mathbf{x}_c = \mathbf{x}_{\text{in}} + (-1)^{n+1}\mathbf{u}_{\mathbf{P}}^0 \times \mathbf{k}, \tag{10}$$

as $\left|\mathbf{u}_{\mathbf{P}}^0\right| = 1$, where $n = 0$ or $n = 1$ and determines the direction of the turn.[2] Similarly, an implicit solution to Equations (4) and (5) is

$$\mathbf{x}_{\text{out}} = \mathbf{x}_c + \frac{(-1)^n}{t_l}\left(\mathbf{x}_I - \mathbf{x}_{\text{out}}\right) \times \mathbf{k}, \tag{11}$$

where Equation (9) has been used. Next, Property 3 and Equation (11) give

$$\mathbf{x}_{\text{out}} \times \mathbf{k} = \mathbf{x}_c \times \mathbf{k} - \frac{(-1)^n}{t_l}\left(\mathbf{x}_I - \mathbf{x}_{\text{out}}\right).$$

Substituting this result into Equation (11) and solving for $\mathbf{x}_{\text{out}}$ leads to an explicit general solution for the exit point:

$$\mathbf{x}_{\text{out}} = \frac{1}{1+t_l^2}\left(\mathbf{x}_I + t_l^2\mathbf{x}_c + (-1)^n t_l\left(\mathbf{x}_I - \mathbf{x}_c\right) \times \mathbf{k}\right). \tag{12}$$

2.2.2 The total time to interception

Equation (9) and Pythagoras' Theorem yield (see Figure 1)

$$t_l = \sqrt{\left|\mathbf{x}_I - \mathbf{x}_c\right|^2 - 1}. \tag{13}$$

This result in conjunction with Equation (6) leads to

$$t_l(T) = \sqrt{(\epsilon T)^2 + 2\beta\epsilon T + \alpha}, \tag{14}$$

as $|\mathbf{u}_{\mathbf{T}}| = 1$, where

$$\alpha = \left|\mathbf{x}_{\mathbf{T}}^0 - \mathbf{x}_c\right|^2 - 1, \tag{15}$$

$$\beta = \mathbf{u}_{\mathbf{T}} \cdot \left(\mathbf{x}_{\mathbf{T}}^0 - \mathbf{x}_c\right). \tag{16}$$

Note that the target is initially outside the pursuer's turning-circle if and only if $\alpha > 0$. The parameter $-\beta$ is the component of the target's heading in the direction of $\mathbf{x}_c$, which follows from the definition of the scalar product.

Equations (12) and (14) reveal that

$$\mathbf{x}_{\text{out}}(T) = \frac{1}{1+t_l^2(T)}\left(\mathbf{x}_I(T) + t_l^2(T)\mathbf{x}_c + (-1)^n t_l(T)\left(\mathbf{x}_I(T) - \mathbf{x}_c\right) \times \mathbf{k}\right). \tag{17}$$

Observe that the only unknown in $\mathbf{x}_{\text{out}}$ is T. As a result the expression for t_c [Equation (8)] can be recast as

$$\cos(T - t_l(T)) = (\mathbf{x}_{\text{in}} - \mathbf{x}_c) \cdot (\mathbf{x}_{\text{out}}(T) - \mathbf{x}_c), \tag{18}$$

since $t_c = T - t_l$. Hence the governing system [Equations (2) to (9)] has been reduced to a single implicit equation for the total time to interception. Equation (18) is valid in an arbitrary Cartesian coordinate system and encompasses both left and right turns. The minimal solution to Equation (18) is sought such that Constraints (C1)–(C3) are satisfied.

[2] If $n = 0$ then the pursuer will turn left, whereas if $n = 1$ then the pursuer will turn right.

2.3 Derivation of the constraints

In this section, mathematical representations of Constraints (C1)–(C3) are derived, and the impact of these constraints on the feasibility of paths to interception is considered one-at-a-time.

(C1): The pursuer cannot reverse its direction on the turning-circle to intercept a target, which can be expressed mathematically as

$$\operatorname{sgn}\big((\mathbf{x}_c - \mathbf{x}_{\text{in}}) \times \mathbf{u}_{\mathbf{P}}^0 \cdot \mathbf{k}\big) = \operatorname{sgn}((\mathbf{x}_c - \mathbf{x}_{\text{out}}) \times (\mathbf{x}_I - \mathbf{x}_{\text{out}}) \cdot \mathbf{k})\,, \tag{19}$$

where

$$\operatorname{sgn}(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0. \end{cases}$$

Equation (19) states that the centre of the turning-circle must remain on the same side of the pursuer as it enters and exits the turning-circle.

Using Property 4 from Section 2.2 together with Equations (11) and (13), it can be shown that

$$(\mathbf{x}_c - \mathbf{x}_{\text{in}}) \times \mathbf{u}_{\mathbf{P}}^0 \cdot \mathbf{k} = (-1)^{n+1},$$

$$(\mathbf{x}_c - \mathbf{x}_{\text{out}}) \times (\mathbf{x}_I - \mathbf{x}_{\text{out}}) \cdot \mathbf{k} = (-1)^{n+1} t_l,$$

and hence $\mathbf{x}_{\text{out}}$ satisfies Constraint (C1) by construction. Furthermore, the turning-circle centred on $\mathbf{x}_c$ with $n = 0$ has an exit point with $n = 0$, and the other turning-circle centred on $\mathbf{x}_c$ with $n = 1$ has an exit point with $n = 1$.

(C2): The pursuer cannot intercept the target inside its turning-circle, that is,

$$|\mathbf{x}_I - \mathbf{x}_c| \geqslant 1,$$

which evaluates to

$$(\epsilon T)^2 + 2\beta\epsilon T + \alpha \geqslant 0, \tag{20}$$

that is, $t_l(T)$ must be a real function [see Equation (14)]. The zeros of $t_l(T)$ are

$$T_L = \frac{-\beta - \sqrt{\beta^2 - \alpha}}{\epsilon}, \qquad T_R = \frac{-\beta + \sqrt{\beta^2 - \alpha}}{\epsilon}. \tag{21}$$

The zeros T_L and T_R are both finite, real and positive when $\alpha > 0$ and $\epsilon > 0$ and $\beta^2 \geqslant \alpha$ and $\beta < 0$, which follows from $|\beta| > \sqrt{\beta^2 - \alpha}$.

It can be seen from Equation (20) that Constraint (C2) is satisfied when $\alpha > 0$ and $\epsilon = 0$, which corresponds to a stationary target that is initially outside the pursuer's turning-circle. Now let $\alpha > 0$ and $0 < \epsilon < 1$. In this case Equations (20) and (21) reveal Constraint (C2) is satisfied when $\beta \geqslant 0$ or $\beta^2 \leqslant \alpha$. The condition $\beta \geqslant 0$ implies the target is heading away from the centre of the turning circle, which follows directly from the definition of β. If $\beta^2 \leqslant \alpha$ is true then the target will never enter the interior of the pursuer's turning-circle, which can be verified as follows. Let $\mathbf{x_T}(t) = \mathbf{x}^0_\mathbf{T} + \epsilon t \mathbf{u_T}$ be the position of the target at time t. The target will never enter the interior of the pursuer's turning-circle if and only if $|\mathbf{x_T}(t) - \mathbf{x}_c| \geqslant 1$ for all t. In this instance Equation (20) will be satisfied and hence $(\epsilon T)^2 + 2\beta\epsilon T + \alpha$ will have at most one real zero, that is, $\beta^2 \leqslant \alpha$. Therefore, if $\alpha > 0$ and $0 < \epsilon < 1$ and $\beta < 0$ and $\beta^2 > \alpha$, the target will enter the pursuer's turning-circle at time T_L and exit the turning-circle at time T_R.

To summarise, Constraint (C2) is satisfied if the target is initially outside the pursuer's turning-circle and will never enter the interior of the turning-circle ($\alpha > 0$ and $\epsilon = 0$ or $\beta \geqslant 0$ or $\beta^2 \leqslant \alpha$). If the target is initially outside the pursuer's turning-circle and will enter the interior of the turning-circle at some time ($\alpha > 0$ and $0 < \epsilon < 1$ and $\beta < 0$ and $\beta^2 > \alpha$), then a solution to Equation (18) that satisfies Constraint (C2) may exist in the domain $0 \leqslant T \leqslant T_L$ or $T \geqslant T_R$.

A discussion of the case where the target is initially inside (or on the boundary of) the pursuer's turning-circle ($\alpha \leqslant 0$) is postponed until Appendix A.3.

(C3): The pursuer may perform at most one complete turn, implying $0 \leqslant t_c \leqslant 2\pi$. However it is only necessary to search for solutions in $0 \leqslant t_c \leqslant \pi$, which can be shown as follows. Given that $\mathbf{x}_{\text{out}}$ is defined on a circle, it is a 2π-periodic function of t_c, as is cosine. Therefore if $\pi \leqslant t_c \leqslant 2\pi$ is a solution of Equation (8) then so is $0 \leqslant 2\pi - t_c \leqslant \pi$, implying the minimal solution will occur in the interval $0 \leqslant t_c \leqslant \pi$.[3]

Given the aforementioned discussion and since $t_c(T) = T - t_l(T)$, a solution to Equation (18) must be restricted to the interval $T_0 \leqslant T \leqslant T_\pi$, where

$$T_0 = \frac{\epsilon\beta + \sqrt{(\epsilon\beta)^2 + (1-\epsilon^2)\alpha}}{1-\epsilon^2}, \tag{22}$$

$$T_\pi = \frac{\epsilon\beta + \pi + \sqrt{t_l^2(\pi) + \epsilon^2(\beta^2 - \alpha)}}{1-\epsilon^2}, \tag{23}$$

$$\overline{T}_\pi = \frac{\epsilon\beta + \pi - \sqrt{t_l^2(\pi) + \epsilon^2(\beta^2 - \alpha)}}{1-\epsilon^2}, \tag{24}$$

which are obtained by solving

$$T_0 - t_l(T_0) = 0, \tag{25}$$

$$T_\pi - t_l(T_\pi) = \pi, \tag{26}$$

$$\overline{T}_\pi - t_l(\overline{T}_\pi) = \pi. \tag{27}$$

Therefore T_0 is the time to interception that corresponds to the pursuer not turning. Likewise, T_π and $\overline{T}_\pi$ are the times to interception that correspond to the pursuer taking π units of time to turn. Refer to Appendix A for a discussion of the validity of these solutions of Equations (25) to (27).

[3] An exception occurs if $0 \leqslant t_c(T) \leqslant \pi$ is the minimal solution of Equation (8) and $(\mathbf{x}_{\text{out}}(T) - \mathbf{x}_{\text{in}}) \cdot \mathbf{u}^0_\mathbf{P} \leqslant 0$, then the minimum time to complete the turn is $2\pi - t_c$. This follows from Figure 1 and the definition of the scalar product; see Section 4.1.2 for more details.

If T_0 and T_π are real numbers, then $T_0 \leqslant T \leqslant T_\pi$ is a necessary condition to satisfy Constraint (C3), however it may not be sufficient. Since $t_c(T) = T - t_l(T)$, if Constraint (C2) is not satisfied then Constraint (C3) cannot be satisfied. Further discussion of the feasibility of times to interception with respect to Constraint (C3) is postponed until Section 3.

It can be seen from Equations (22) to (24) that there is a singularity at $\epsilon = 1$. Although solutions of Equations (2) to (9) exist for the unusual case of $\epsilon \geqslant 1$, these solutions have a different form to those obtained for $0 \leqslant \epsilon < 1$. Since the case $0 \leqslant \epsilon < 1$ is of greater practical interest, for the reminder of the paper it is assumed that $0 \leqslant \epsilon < 1$.

3 Feasible times to interception

Feasible times to interception are defined to be those times which simultaneously satisfy Constraints (C1)–(C3). However, there is room for confusion here as Equation (18) is both an additional constraint and the objective function. This is because a solution to Equation (18) may not exist from the set of feasible times to interception (called the feasible region), and hence Equation (18) acts as an additional constraint. If solutions to Equation (18) do exist from the feasible region, then Equation (18) becomes the objective function and the minimum solution is chosen.

A rigorous analysis of Constraints (C1)–(C3) has been performed in Appendix A. The results from this appendix and Section 2.3 will now be utilised to determine the feasible region. To simplify the discussion, only the case where the target is initially outside the pursuer's turning-circle, that is, $\alpha > 0$, will be considered here; the case when $\alpha \leqslant 0$ is analyzed in Appendix A.3.

In Section 2.3 it was shown that Constraint (C1) is satisfied by construction, and (if $\alpha > 0$) there are two main cases effecting the feasibility of times to interception with respect to Constraints (C2) and (C3). These cases are:

- the target will never enter the interior of the pursuer's turning-circle ($\alpha > 0$ and $\epsilon = 0$ or $\beta \geqslant 0$ or $\beta^2 \leqslant \alpha$), and;
- the target will enter the interior of the pursuer's turning-circle at some time ($\alpha > 0$ and $0 < \epsilon < 1$ and $\beta < 0$ and $\beta^2 > \alpha$).

Hence the parameters effecting the feasibility of times to interception are $\alpha = \left|\mathbf{x}_{\mathbf{T}}^0 - \mathbf{x}_c\right|^2 - 1$, $\beta = \mathbf{u}_{\mathbf{T}} \cdot \left(\mathbf{x}_{\mathbf{T}}^0 - \mathbf{x}_c\right)$ and $\epsilon = C_{\mathbf{T}}/C_{\mathbf{P}}$. Furthermore, when $\alpha > 0$, it is natural to partition the parameter space into the disjoint sets

$$\left\{(\alpha, \beta, \epsilon) \mid \epsilon = 0 \text{ or } \beta \geqslant 0 \text{ or } \beta^2 \leqslant \alpha\right\},$$

and

$$\left\{(\alpha, \beta, \epsilon) \mid 0 < \epsilon < 1 \text{ and } \beta < 0 \text{ and } \beta^2 > \alpha\right\}.$$

Proposition 7 of Appendix A.2 states that if $\alpha > 0$ and $\epsilon = 0$ or $\beta \geqslant 0$ or $\beta^2 \leqslant \alpha$, then feasible times to interception T occur in the interval $T_0 \leqslant T \leqslant T_\pi$. In this instance, the target will never enter the interior of the pursuer's turning-circle and so the pursuer is free to perform a complete turn before attempting an intercept. In fact, it is proven in Proposition 8 of Appendix A.2 that the pursuer will have at least one opportunity to intercept the target.

The other case, when $\alpha > 0$ and $0 < \epsilon < 1$ and $\beta < 0$ and $\beta^2 > \alpha$, is more complex. In this situation there are four conditions effecting the feasibility of times to interception that lead to four different feasible regions, which are presented in Table 1. The results in this table are proven in Propositions 9 to 12 of Appendix A.2. These propositions establish the shape of $t_c(T)$ which is shown in Figure 2(a)–(d), where the four feasible regions in Table 1 can also be obtained by inspection. Recall that in all four cases the target will

Case	Condition	Feasible Region
1	$T_R \leqslant \pi$	$T_0 \leqslant T \leqslant T_L$ and $T_R \leqslant T \leqslant T_\pi$
2	$T_L < \pi < T_R$ and T_π is real	$T_0 \leqslant T \leqslant T_L$ and $\overline{T}_\pi \leqslant T \leqslant T_\pi$
3	$T_L < \pi < T_R$ and T_π is complex	$T_0 \leqslant T \leqslant T_L$
4	$T_L \geqslant \pi$	$T_0 \leqslant T \leqslant T_\pi$

Table 1: *The four cases effecting the feasibility of times to interception when the target will enter the interior of the pursuer's turning-circle at some time ($\alpha > 0$ and $0 < \epsilon < 1$ and $\beta < 0$ and $\beta^2 > \alpha$).*

enter the interior of the pursuer's turning-circle at some time. Physical meaning can then be attributed to the four cases as follows:[4]

Case 1: The target will exit the pursuer's turning-circle before time $t = 2\pi$, and the pursuer can feasibly intercept the target after it exits the turning-circle. Equivalently, the target is moving fast enough for the pursuer to attempt an intercept immediately after the target leaves the turning-circle.

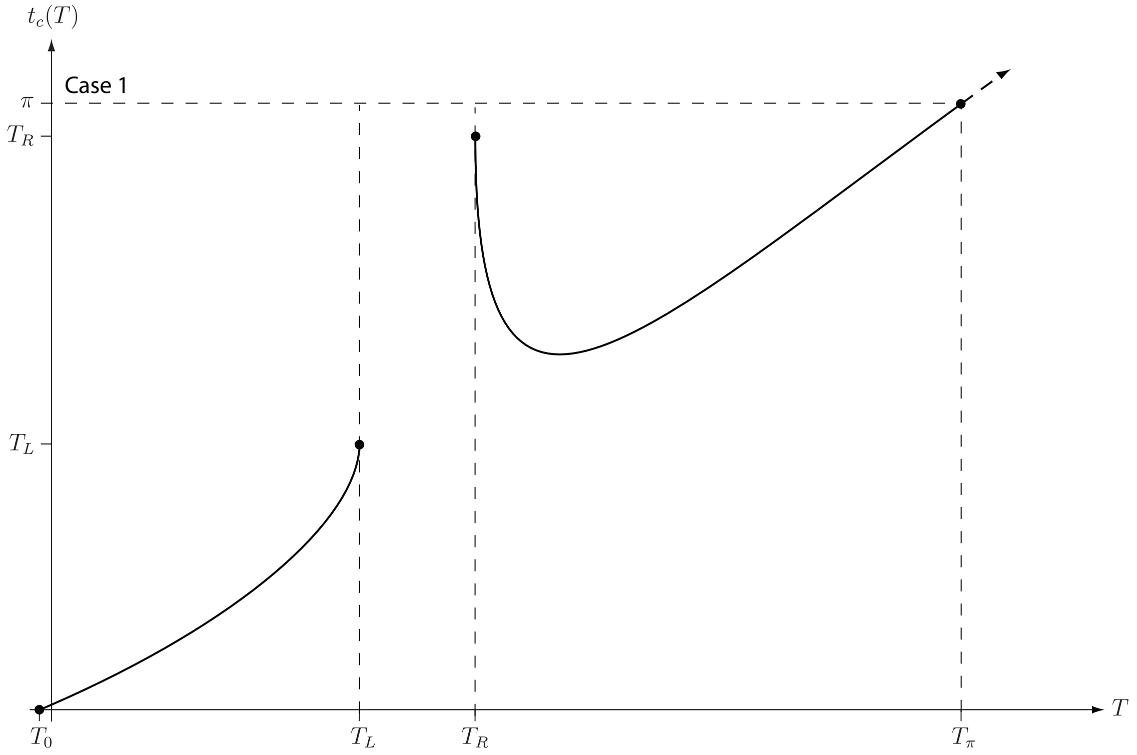
Case 2: The target will exit the pursuer's turning-circle before $t = 2\pi$, however the pursuer cannot feasibly intercept the target after it exits the turning-circle until $t = \overline{T}_\pi$. Equivalently, the target is moving fast enough for the pursuer to attempt an intercept $\overline{T}_\pi - T_R$ units of time after the target leaves the turning-circle.

Case 3: The target will not exit the pursuer's turning-circle before $t = 2\pi$, and hence the pursuer cannot feasibly intercept the target after it exits the turning-circle. Equivalently, the target has enough speed to enter the pursuer's turning-circle, but is moving too slowly to leave the turning-circle before the pursuer has performed a complete turn.

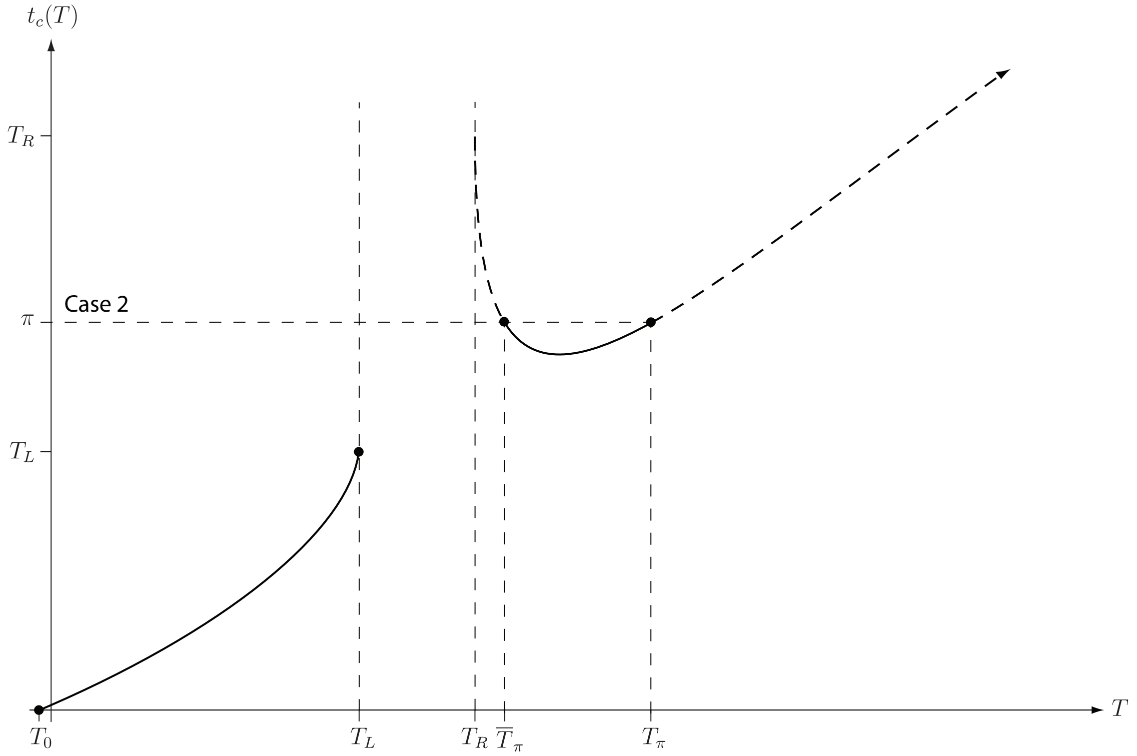
Case 4: The target will not enter the pursuer's turning-circle before $t = 2\pi$, and hence the pursuer can feasibly perform a complete turn before intercepting the target. Equivalently, the target is moving so slowly, and/or it is so far away from the pursuer's turning-circle, that it will not enter the turning-circle before the pursuer has performed a complete turn.

The above cases only describe the feasibility of times to interception with respect to Constraints (C1)–(C3), and feasibility does not imply that the pursuer can intercept the target. However, it is proven in Proposition 9 of Appendix A.2 that if $T_L \geqslant \pi$ (Case 4), then the pursuer will have at least one opportunity to intercept the target.

[4] The speed of the target is relative to the speed of the pursuer.

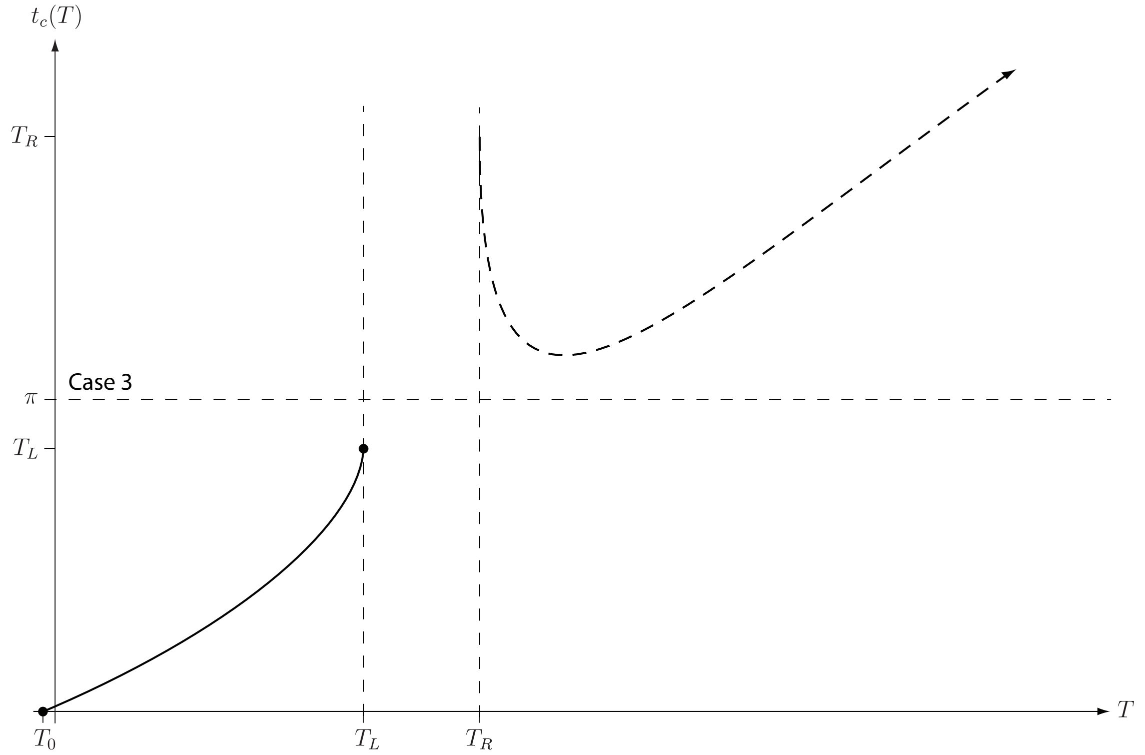


(a) If $T_R \leqslant \pi$, then the feasible region is $T_0 \leqslant T \leqslant T_L$ and $T_R \leqslant T \leqslant T_\pi$. (See Proposition 10 in Appendix A.2 for a proof of this statement.)

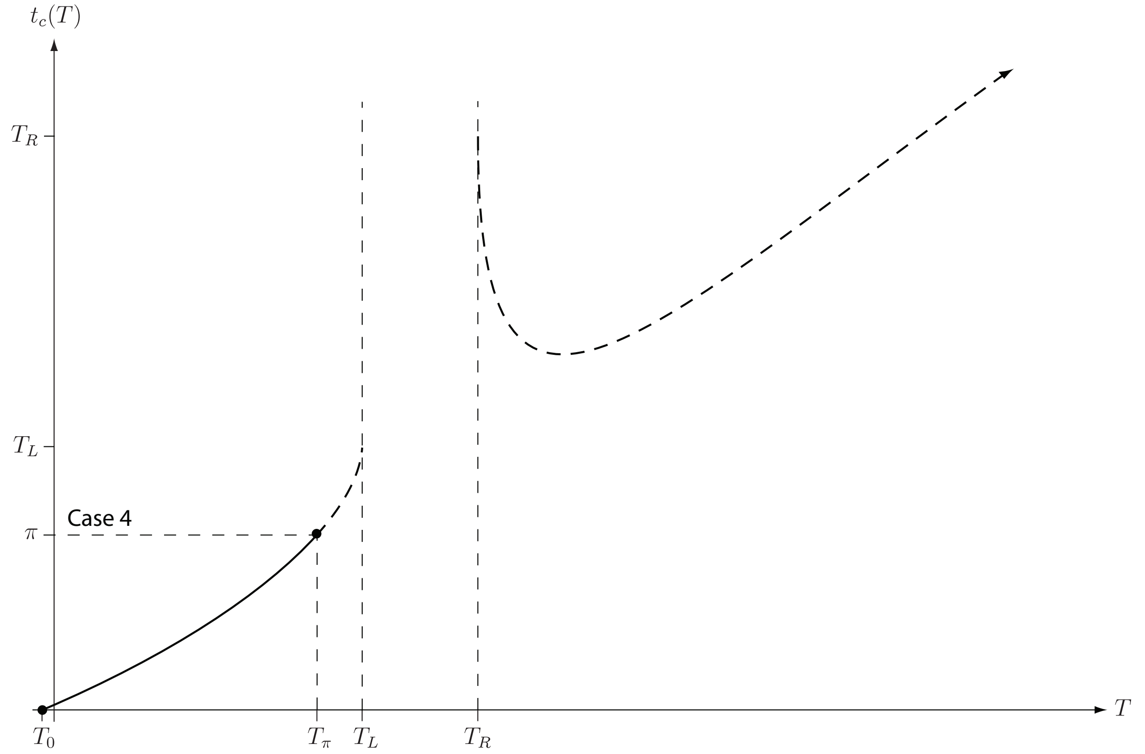


(b) If $T_L < \pi < T_R$ and T_π is real, then the feasible region is $T_0 \leqslant T \leqslant T_L$ and $\overline{T}_\pi \leqslant T \leqslant T_\pi$. (See Proposition 12 in Appendix A.2 for a proof of this statement.)

Figure 2: *The time for the pursuer to complete its turn $t_c(T) = T - t_l(T)$ versus the total time to interception T, for $\alpha > 0$ and $0 < \epsilon < 1$ and $\beta < 0$ and $\beta^2 > \alpha$ (Cases 1 and 2). The part of the graph that is displayed as a solid line is the feasible region.*



(c) If $T_L < \pi < T_R$ and T_π is complex, then the feasible region is $T_0 \leqslant T \leqslant T_L$. (See Proposition 11 in Appendix A.2 for a proof of this statement.)



(d) If $T_L \geqslant \pi$, then the feasible region is $T_0 \leqslant T \leqslant T_\pi$. (See Proposition 9 in Appendix A.2 for a proof of this statement.)

Figure 2: *The time for the pursuer to complete its turn $t_c(T) = T - t_l(T)$ versus the total time to interception T, for $\alpha > 0$ and $0 < \epsilon < 1$ and $\beta < 0$ and $\beta^2 > \alpha$ (Cases 3 and 4). The part of the graph that is displayed as a solid line is the feasible region.*

4 The *Interception* algorithm

The results obtained in Section 3 have been utilised to develop an algorithm that can be used to systematically search the feasible region for the minimum time to interception. The algorithm, called *Interception*, is shown in Figure 3. *A description of* Interception *using words, rather than symbols, is presented in Appendix B.*

To simplify the discussion, only the case where the target is initially outside the pursuer's turning-circle, that is, $\alpha > 0$, will be considered in this section; the *Interception* algorithm for the case when $\alpha \leqslant 0$ is derived in Appendix A.3 and displayed in Figure C1(a)–(b).

4.1 Implementing *Interception*

In this section, the practicalities of implementing *Interception* are discussed.

4.1.1 Convergence of the root-finding method

The *Interception* algorithm can be used to systematically search the feasible region for the minimum time to interception, that is, the minimum feasible solution of Equation (18). However, the reliability of *Interception* to converge to the minimum time to interception is limited by the root-finding method used to solve Equation (18). For instance, the root-finding method may fail to converge, even when a solution does exist, or it may not converge to the minimum solution. These issues can be alleviated as follows.

Suppose a solution to Equation (18) is sought in the interval $L \leqslant \widetilde{T} \leqslant R$. Then choose δ_1 and ϕ such that $\delta_1 > 1$ and $0 < \phi < 1$. Since the minimum solution is sought, begin the search at $L + (R - L)/\delta_1$. If the root-finding method fails to converge to a solution in the interval $L \leqslant \widetilde{T} \leqslant R$, then set $\delta_2 = \phi\delta_1$ and search again, this time starting at $L + (R - L)/\delta_2$. Continue in this manner, starting at $L + (R - L)/\delta_N$ for the Nth attempt ($\delta_N = \phi^{N-1}\delta_1$), until the root-finding method converges to a solution of Equation (18) in the interval $L \leqslant \widetilde{T} \leqslant R$ or $\delta_N \leqslant 1$. If a solution has still not been found, depending on which root-finding method is being used, it may be necessary to repeat this procedure starting the search at $R - (R - L)/\delta_1$. By changing the parameters δ_1 and ϕ the balance between reliability and computational performance may be adjusted.

Root-finding methods for nonlinear equations will not be discussed further in this paper as there are a multitude of text books on numerical analysis that cover the topic; for example, see Atkinson [1989].

4.1.2 The minimum feasible time to interception

Although Constraint (C3) implies that $0 \leqslant t_c \leqslant 2\pi$, it is only necessary to search for a solution to Equation (18) in $0 \leqslant t_c \leqslant \pi$, as discussed in Section 2.3. However, the resulting time to interception will not equal the true minimum time to interception if the pursuer performs a turn of more than π radians. This does not cause any problems,

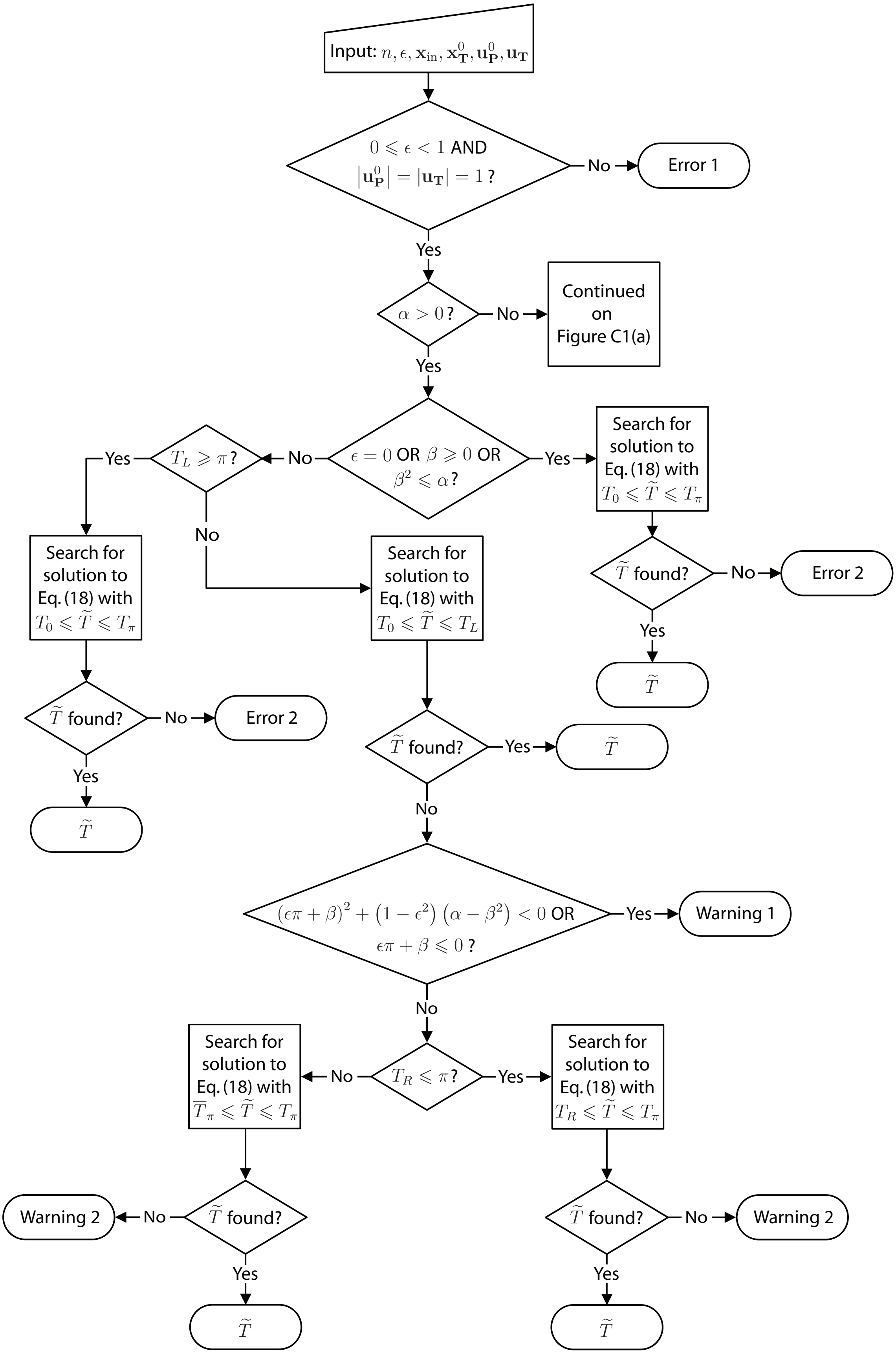

Input: $n, \epsilon, \mathbf{x}_{\text{in}}, \mathbf{x}_{\mathbf{T}}^{0}, \mathbf{u}_{\mathbf{P}}^{0}, \mathbf{u}_{\mathbf{T}}$
$0 \leqslant \epsilon < 1$ AND $\left|\mathbf{u}_{\mathbf{P}}^{0}\right| = \left|\mathbf{u}_{\mathbf{T}}\right| = 1$?
No
Error 1
Yes
$\alpha > 0$?
No
Continued on Figure C1(a)
Yes
$\epsilon = 0$ OR $\beta \geqslant 0$ OR $\beta^2 \leqslant \alpha$?
Yes
Search for solution to Eq. (18) with $T_0 \leqslant \widetilde{T} \leqslant T_\pi$
$\widetilde{T}$ found?
No
Error 2
Yes
$\widetilde{T}$
No
$T_L \geqslant \pi$?
Yes
Search for solution to Eq. (18) with $T_0 \leqslant \widetilde{T} \leqslant T_\pi$
$\widetilde{T}$ found?
No
Error 2
Yes
$\widetilde{T}$
No
Search for solution to Eq. (18) with $T_0 \leqslant \widetilde{T} \leqslant T_L$
$\widetilde{T}$ found?
Yes
$\widetilde{T}$
No
$(\epsilon\pi + \beta)^2 + (1 - \epsilon^2)(\alpha - \beta^2) < 0$ OR $\epsilon\pi + \beta \leqslant 0$?
Yes
Warning 1
No
$T_R \leqslant \pi$?
No
Search for solution to Eq. (18) with $T_\pi \leqslant \widetilde{T} \leqslant T_\pi$
$\widetilde{T}$ found?
No
Warning 2
Yes
$\widetilde{T}$
Yes
Search for solution to Eq. (18) with $T_R \leqslant \widetilde{T} \leqslant T_\pi$
$\widetilde{T}$ found?
No
Warning 2
Yes
$\widetilde{T}$

Figure 3: *The Interception algorithm for determining feasible times to interception $\widetilde{T}$; the case when $\alpha \leqslant 0$ is continued on Figure C1(a). A description of* Interception *using words, rather than symbols, is presented in Appendix B.*

because the true minimum time to interception can be easily obtained from the solution of Equation (18) as follows.

The true minimum time for the pursuer to complete its turn is

$$\widetilde{t}_c(\widetilde{T}) = \begin{cases} \widetilde{T} - t_l(\widetilde{T}), & (\mathbf{x}_{\text{out}}(\widetilde{T}) - \mathbf{x}_{\text{in}}) \cdot \mathbf{u}^0_{\mathbf{P}} > 0 \\ 2\pi - (\widetilde{T} - t_l(\widetilde{T})), & (\mathbf{x}_{\text{out}}(\widetilde{T}) - \mathbf{x}_{\text{in}}) \cdot \mathbf{u}^0_{\mathbf{P}} \leqslant 0, \end{cases} \tag{28}$$

which follows from Figure 1 and the definition of the scalar product. The true minimum feasible time to interception is then given by

$$T = \widetilde{t}_c(\widetilde{T}) + t_l(\widetilde{T}), \tag{29}$$

where $\widetilde{T}$ is the minimum feasible solution of Equation (18) returned by *Interception*; see Figure 3.

4.1.3 Errors and warnings

When *Interception* is executed, a number of error and warning messages may be returned that correspond to physical events.[5] These messages are described below:

Error 1 is returned if

- $\epsilon < 0$ or $\epsilon \geqslant 1$: $\epsilon < 0$ is not physically possible and $\epsilon \geqslant 1$ implies the target's speed is greater than or equal to the speed of the pursuer, or;
- $|\mathbf{u}^0_{\mathbf{P}}| \neq 1$ or $|\mathbf{u}_{\mathbf{T}}| \neq 1$: the headings have been scaled such that $|\mathbf{u}^0_{\mathbf{P}}| = |\mathbf{u}_{\mathbf{T}}| = 1$.[6]

Error 2 is returned if the root-finding method used to solve Equation (18) fails to converge, when a solution is known to exist; see Propositions 8 and 9 from Appendix A.2.

Warning 1 is returned if the root-finding method used to solve Equation (18) fails to converge. In this instance, the pursuer is unable to intercept the target before it enters the turning-circle (if $\alpha > 0$), and the target is moving too slowly to leave the turning-circle before the pursuer has performed a complete turn; see Case 3 from Section 3.

Warning 2 is returned if the root-finding method used to solve Equation (18) fails to converge. In this instance, the pursuer is unable to intercept the target before it enters the turning-circle (if $\alpha > 0$), and the target is moving too slowly to leave the turning-circle in time for the pursuer to attempt an intercept; see Cases 1 and 2 from Section 3.

[5] Error 2 does not correspond to a physical event.

[6] This error is irrelevant if *Interception* is implemented using dimensional quantities.

4.1.4 A modified definition of α

The overall behaviour of *Interception* is determined by the sign of α (see Figure 3). It has been found that *Interception* is very sensitive to numerical error if $\left|\mathbf{x}_\mathbf{T}^0 - \mathbf{x}_c\right| \approx 1$, which stems from the definition of α:

$$\alpha = \left|\mathbf{x}_\mathbf{T}^0 - \mathbf{x}_c\right|^2 - 1 = \left(\left|\mathbf{x}_\mathbf{T}^0 - \mathbf{x}_c\right| - 1\right)\left(\left|\mathbf{x}_\mathbf{T}^0 - \mathbf{x}_c\right| + 1\right) .$$

It can be demonstrated that the squared term introduces an additional numerical error if $\left|\mathbf{x}_\mathbf{T}^0 - \mathbf{x}_c\right| \approx 1$, that is, if $\left|\mathbf{x}_\mathbf{T}^0 - \mathbf{x}_c\right|$ numerically equals unity, α will not numerically equal zero. The following modified definition of α alleviates this problem:

$$\alpha = \begin{cases} \left|\mathbf{x}_\mathbf{T}^0 - \mathbf{x}_c\right|^2 - 1, & \left|\left|\mathbf{x}_\mathbf{T}^0 - \mathbf{x}_c\right| - 1\right| > \delta_{tol} \\ 0, & \left|\left|\mathbf{x}_\mathbf{T}^0 - \mathbf{x}_c\right| - 1\right| \leqslant \delta_{tol} \end{cases}$$

where $0 < \delta_{tol} \ll 1$ is a numerical tolerance.

4.1.5 Polar coordinates

It may be useful to express the points $\mathbf{x}_{\text{in}}$ and $\mathbf{x}_{\text{out}}$ in polar coordinates centred on $\mathbf{x}_c$. A difficulty here is, standard trigonometric methods for transforming points in Cartesian coordinates into polar coordinates, may return points in the wrong quadrants. A method to obtain expressions for $\mathbf{x}_{\text{in}}$ and $\mathbf{x}_{\text{out}}$ in polar coordinates centred on $\mathbf{x}_c$, such that the resulting points are in the correct quadrants, is presented in Appendix E.

4.2 An implementation of *Interception*

For the purpose of validation, the *Interception* algorithm has been implemented as a Mathematica[7] package, called `TurningCircle.m`, that displays the minimum feasible path to interception. The code for `TurningCircle.m` is included in Appendix D.

Whether or not `TurningCircle.m` returns the minimum path to interception depends on the reliability of the root-finding method employed in *Interception* to converge to the minimum solution of Equation (18). The technique described in Section 4.1.1 for addressing this issue has been employed in `TurningCircle.m`. It was found by trial-and-error that setting $\delta_1 = 70000$ and $\phi = 0.3$ produced faultless results for $0 \leqslant \epsilon \leqslant 0.9999$, regardless of the other inputs.[8] With these parameter values, `TurningCircle.m` will terminate after a maximum of 20 attempts (10 starting from the left endpoint plus 10 starting from the right endpoint).[9]

[7] See `http://www.wolfram.com/` for more information.

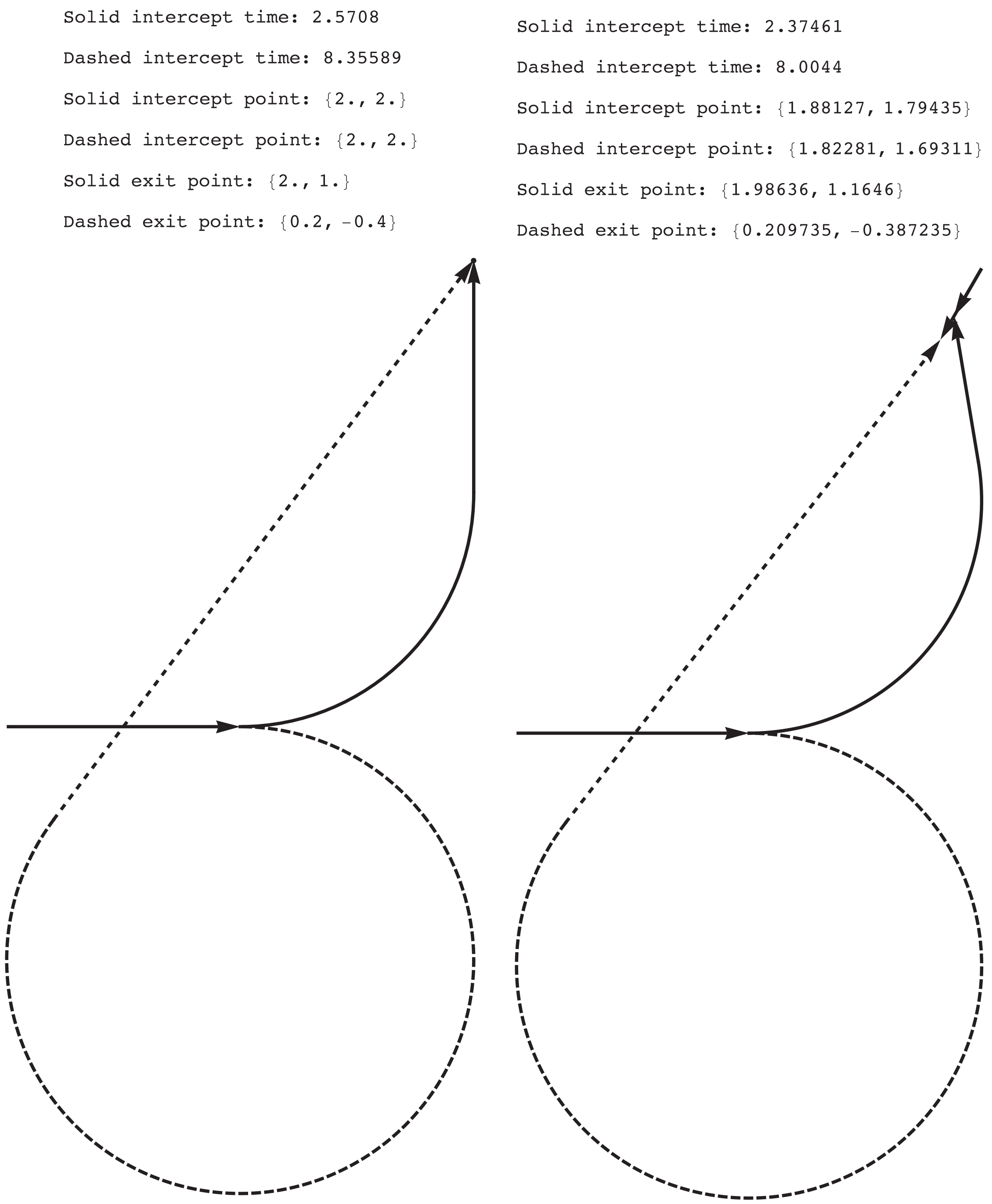
[8] `TurningCircle.m` was not subjected to rigorous testing. However, after numerous comparisons with results obtained graphically from Equation (18), `TurningCircle.m` returned no false outcomes.

[9] To obtain an indication of execution time, `TurningCircle.m` was run 100 times for each of the parameter values used to generate Figure 4(a)–(f) (with $\delta_1 = 70000$ and $\phi = 0.3$). The 95% confidence interval for the mean execution time in CPU seconds is $[0.09, 0.40]$. `TurningCircle.m` was run using Mathematica 6.0.2.1 on an Apple iMac running Mac OS X 10.4.11 with a 2 GHz PowerPC G5 CPU and 1 GB of RAM.

4.2.1 Examples of the turning-radius effect

Six examples of the output generated by `TurningCircle.m` are displayed in Figure 4(a)–(f). The target's speed increases (relative to the pursuer's speed) from Figure 4(a) to Figure 4(f). These figures illustrate the effect of turning radius as they encompass all of the cases discussed in Section 3.

The case of a stationary target is shown in Figure 4(a), where the pursuer can turn either left or right to intercept the target. This is always true for a stationary target, regardless of the initial heading and speed of the pursuer, provided the target is not inside one of the pursuer's turning-circles (see Proposition 8 of Appendix A.2). In Figure 4(b) the target is moving slowly enough for the pursuer to intercept the target before it enters either turning-circle. In Figure 4(c) Warning 1 (see Section 4.1.3) has been returned, and consequently the pursuer must turn away from the target to perform an intercept. Warning 2 (see Section 4.1.3) has been returned in Figure 4(d) and, although the target has greater speed, the pursuer must still turn away from the target to perform an intercept. In Figures 4(e) and 4(f), the target has sufficient speed for the pursuer to turn towards the target to perform an intercept (the solid path), however, this results in the pursuer being required to chase the target. It is interesting to observe that increasing the relative speed of the target from $\epsilon = 0.95$ to $\epsilon = 0.9999$ leads to a reduction in the time to interception from 8.529 to 7.073, respectively, despite the pursuer being required to chase the target. Although counterintuitive, this phenomena occurs because as the target's speed increases, the time taken for it to exit the turning-circle decreases, thus enabling an intercept to occur earlier.



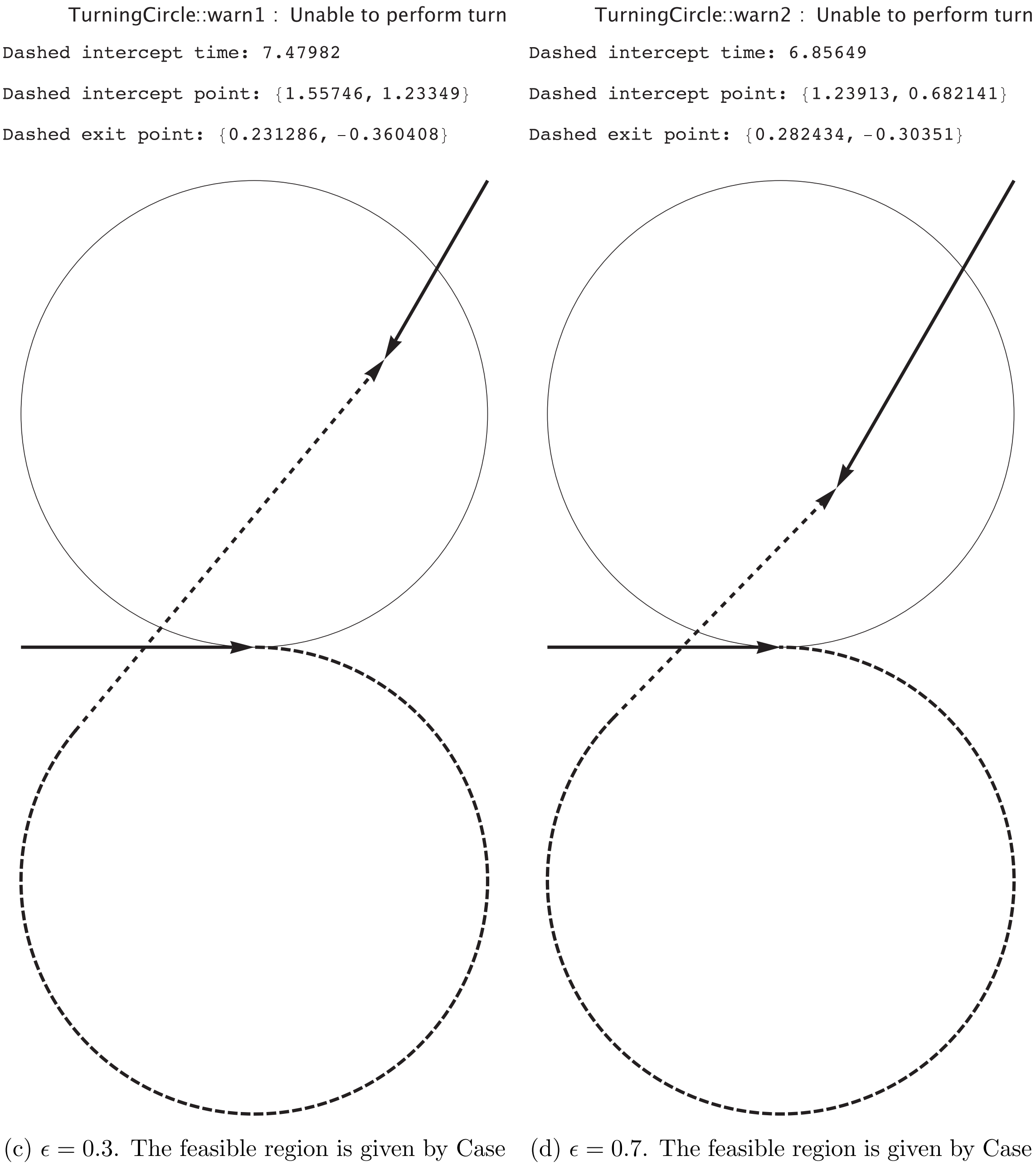
```
Solid intercept time: 2.5708

Dashed intercept time: 8.35589

Solid intercept point: {2., 2.}

Dashed intercept point: {2., 2.}

Solid exit point: {2., 1.}

Dashed exit point: {0.2, -0.4}
```

(a) $\epsilon = 0$, which corresponds to a stationary target.

```
Solid intercept time: 2.37461

Dashed intercept time: 8.0044

Solid intercept point: {1.88127, 1.79435}

Dashed intercept point: {1.82281, 1.69311}

Solid exit point: {1.98636, 1.1646}

Dashed exit point: {0.209735, -0.387235}
```

(b) $\epsilon = 0.1$. The feasible region is given by Case 4 for both turns.

Figure 4: *Output from* `TurningCircle.m`*, which is an implementation of the* Interception *algorithm as a Mathematica package. The pursuer's path to interception is displayed as a solid line for a left turn, and as a dashed line for a right turn. The inputs are:* $\mathbf{x}_{\text{in}} = (1, 0)$, $\mathbf{x}_{\mathbf{T}}^0 = (2, 2)$, $\mathbf{u}_{\mathbf{P}}^0 = (1, 0)$, *and* $\mathbf{u}_{\mathbf{T}} = -(\sqrt{0.5/2}, \sqrt{1.5/2})$, *for different values of* $\epsilon = C_{\mathbf{T}}/C_{\mathbf{P}}$ *(*$\epsilon = 0$ *and* $\epsilon = 0.1$*).*



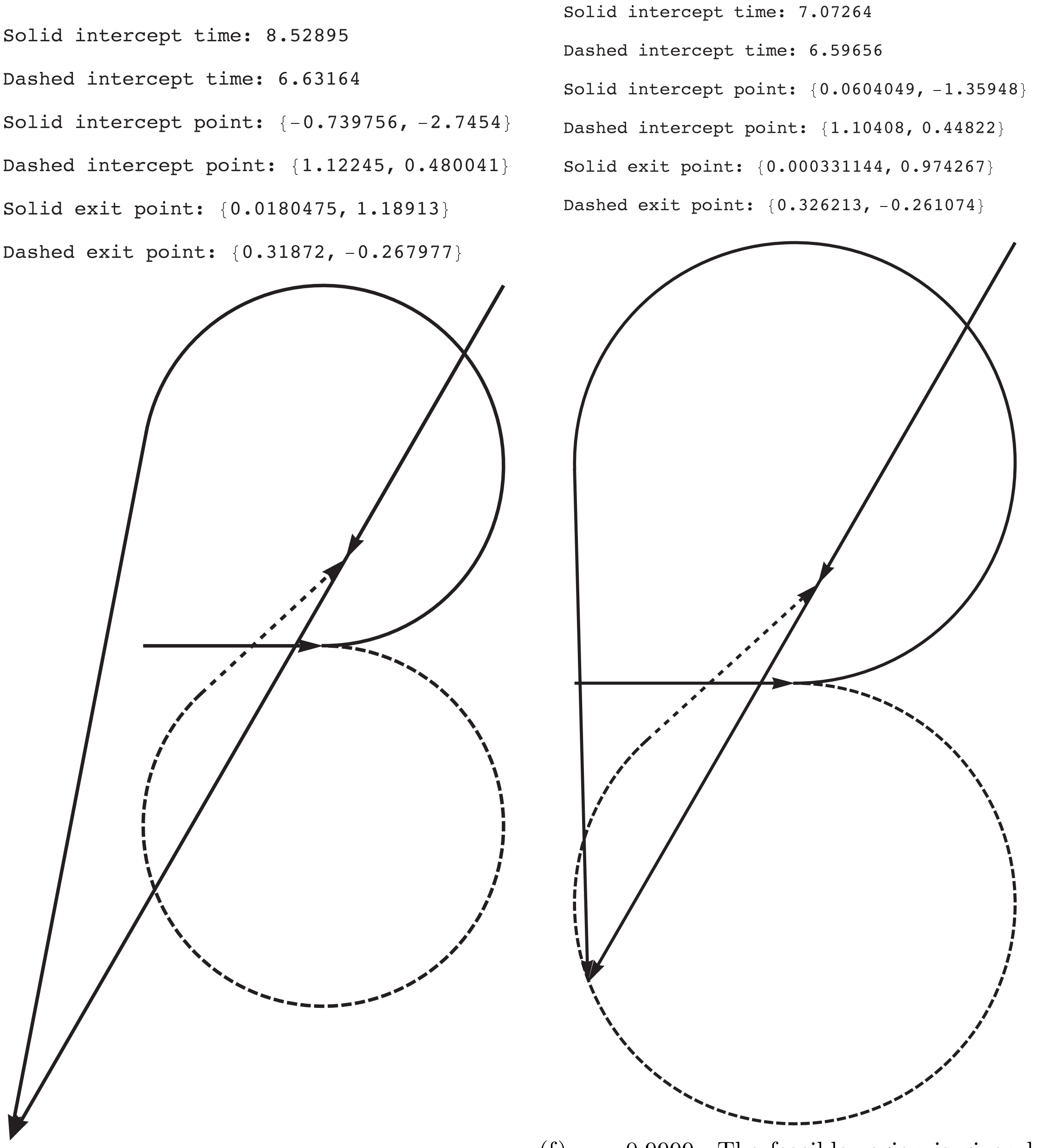
```
TurningCircle::warn1 : Unable to perform turn
Dashed intercept time: 7.47982
Dashed intercept point: {1.55746, 1.23349}
Dashed exit point: {0.231286, -0.360408}
```

```
TurningCircle::warn2 : Unable to perform turn
Dashed intercept time: 6.85649
Dashed intercept point: {1.23913, 0.682141}
Dashed exit point: {0.282434, -0.30351}
```

(c) $\epsilon = 0.3$. The feasible region is given by Case 3 for the left turn, and Case 4 for the right turn.

(d) $\epsilon = 0.7$. The feasible region is given by Case 2 for the left turn, and Case 4 for the right turn.

Figure 4: *Output from* `TurningCircle.m`*, which is an implementation of the* Interception *algorithm as a Mathematica package. The pursuer's path to interception is displayed as a solid line for a left turn, and as a dashed line for a right turn. The inputs are:* $\mathbf{x}_{\text{in}} = (1, 0)$*,* $\mathbf{x}^0_{\mathbf{T}} = (2, 2)$*,* $\mathbf{u}^0_{\mathbf{P}} = (1, 0)$*, and* $\mathbf{u}_{\mathbf{T}} = -(\sqrt{0.5/2}, \sqrt{1.5/2})$*, for different values of* $\epsilon = C_{\mathbf{T}}/C_{\mathbf{P}}$ *(*$\epsilon = 0.3$ *and* $\epsilon = 0.7$*).*



```
Solid intercept time: 8.52895

Dashed intercept time: 6.63164

Solid intercept point: {-0.739756, -2.7454}

Dashed intercept point: {1.12245, 0.480041}

Solid exit point: {0.0180475, 1.18913}

Dashed exit point: {0.31872, -0.267977}
```

(e) $\epsilon = 0.95$. The feasible region is given by Case 1 for the left turn, and Case 3 for the right turn.

```
Solid intercept time: 7.07264

Dashed intercept time: 6.59656

Solid intercept point: {0.0604049, -1.35948}

Dashed intercept point: {1.10408, 0.44822}

Solid exit point: {0.000331144, 0.974267}

Dashed exit point: {0.326213, -0.261074}
```

(f) $\epsilon = 0.9999$. The feasible region is given by Case 1 for the left turn, and Case 2 for the right turn.

Figure 4: *Output from `TurningCircle.m`, which is an implementation of the* Interception *algorithm as a Mathematica package. The pursuer's path to interception is displayed as a solid line for a left turn, and as a dashed line for a right turn. The inputs are:* $\mathbf{x}_{\text{in}} = (1, 0)$, $\mathbf{x}^0_{\mathbf{T}} = (2, 2)$, $\mathbf{u}^0_{\mathbf{P}} = (1, 0)$, *and* $\mathbf{u}_{\mathbf{T}} = -(\sqrt{0.5/2}, \sqrt{1.5/2})$, *for different values of* $\epsilon = C_{\mathbf{T}}/C_{\mathbf{P}}$ *(*$\epsilon = 0.95$ *and* $\epsilon = 0.9999$*).*

5 Incorporating classification range

In maritime surveillance operations, aircrew can classify a ship if the aircraft is within a distance of $\bar{r}_{cl}$ from the ship, where $\bar{r}_{cl}$ is the aircraft's classification range that depends on the aircraft's sensor suite, weather, sea state, *et cetera*.[10] In this section, three methods are proposed for incorporating classification range into the present model: an exact method, a heuristic method, and a method that includes an angle of approach.

5.1 Exact method

The pursuer's classification range can be incorporated into the present model by transforming the interception point into the classification point. Let T_{cl} be the time to classification and

$$\mathbf{x_T}(T_{cl}) = \mathbf{x}_\mathbf{T}^0 + \epsilon T_{cl}\mathbf{u_T}, \tag{30}$$

the position of the target at the time of classification. The classification point $\mathbf{x}_{cl}$ is defined to be at a distance of $\bar{r}_{cl}$ from $\mathbf{x_T}(T_{cl})$ in the direction of $\mathbf{x}_{\text{out}}(T_{cl}) - \mathbf{x_T}(T_{cl})$; that is, the classification point results from the most direct path to classification. It follows that $\mathbf{x}_{cl}$ is given by

$$\mathbf{x}_{cl}(T_{cl}) = \mathbf{x_T}(T_{cl}) + \frac{\bar{r}_{cl}}{|\mathbf{x}_{\text{out}}(T_{cl}) - \mathbf{x_T}(T_{cl})|}\left(\mathbf{x}_{\text{out}}(T_{cl}) - \mathbf{x_T}(T_{cl})\right). \tag{31}$$

To determine T_{cl} it is necessary to replace the interception point $\mathbf{x}_I$ with $\mathbf{x}_{cl}$ in Equations (13) and (17). As a consequence, Equation (17) becomes an implicit expression for $\mathbf{x}_{\text{out}}(T_{cl})$, as $t_l(T_{cl})$ and $\mathbf{x}_{cl}(T_{cl})$ are also functions of $\mathbf{x}_{\text{out}}(T_{cl})$. Hence the following system of equations must be solved to determine the minimum time to classification:

$$t_l(T_{cl}) = \sqrt{|\mathbf{x}_{cl}(T_{cl}) - \mathbf{x}_c|^2 - 1}, \tag{32}$$

$$\mathbf{x}_{\text{out}}(T_{cl}) = \frac{1}{1 + t_l^2(T_{cl})}\big(\mathbf{x}_{cl}(T_{cl}) + t_l^2(T_{cl})\mathbf{x}_c + (-1)^n t_l(T_{cl})\left(\mathbf{x}_{cl}(T_{cl}) - \mathbf{x}_c\right) \times \mathbf{k}\big), \tag{33}$$

$$\cos(T_{cl} - t_l(T_{cl})) = (\mathbf{x}_{\text{in}} - \mathbf{x}_c) \cdot (\mathbf{x}_{\text{out}}(T_{cl}) - \mathbf{x}_c), \tag{34}$$

where $\mathbf{x}_{cl}$ is given by Equation (31).

Since Equation (33) is an implicit expression for $\mathbf{x}_{\text{out}}(T_{cl})$, nearly all of the analysis in Section 3 does not apply to Equations (31) to (34), and so the *Interception* algorithm cannot be easily modified to accommodate this system. Furthermore, the analysis of Equations (31) to (34) is far from straightforward, making Constraints (C1)–(C3) difficult to apply.

5.2 Heuristic method

A heuristic method that accounts for the pursuer's classification range and returns feasible times to classification, can be obtained by initially determining the minimum time to

[10] The classification range in this section refers to the dimensionless quantity $\bar{r}_{cl} = r_{cl}/r_c$, where r_{cl} is the dimensional classification range.

interception T using Equation (29) and *Interception*. An approximate classification point is then derived by starting at $\mathbf{x}_I(T)$ and moving a distance of $\bar{r}_{cl}$ in the direction of $\mathbf{x}_{\text{out}}(T) - \mathbf{x}_I(T)$. The resulting classification point is given by

$$\mathbf{x}_{cl} = \begin{cases} \mathbf{x}_I(T) + \dfrac{\bar{r}_{cl}}{t_l(T)} \left(\mathbf{x}_{\text{out}}(T) - \mathbf{x}_I(T)\right), & \bar{r}_{cl} < t_l(T) \\ \mathbf{x}_{\text{out}}(T), & \bar{r}_{cl} \geqslant t_l(T), \end{cases}$$

and the time to classification is simply[11]

$$T_{cl} = \begin{cases} T - \bar{r}_{cl}, & \bar{r}_{cl} < t_l(T) \\ t_c(T), & \bar{r}_{cl} \geqslant t_l(T). \end{cases} \tag{35}$$

Observe that if $\bar{r}_{cl} \geqslant t_l(T) = |\mathbf{x}_{\text{out}}(T) - \mathbf{x}_I(T)|$ then the target is inside the pursuer's classification range when the pursuer exits its turn, that is, at time $t_c(T)$.

Equation (35) yields a feasible time to classification. However, it will not necessarily return the minimum time to classification, which can be demonstrated as follows. Since the classification point is closer to the pursuer than the interception point, the pursuer's classification range acts as if to increase the speed of the pursuer. Consequently the pursuer may be able to classify the target, but unable to perform an intercept. In this instance the heuristic will fail because the approximate classification point is derived from the interception point.

5.3 Angle of approach

The pursuer's sensor performance or its path can usually be optimised by classifying the target at a specified angle of approach η, which is relative to the target [Mercer et al. 2008]; see Figure 5. In this case, the classification point becomes[12]

$$\mathbf{x}_{cl}(T_{cl}) = \mathbf{x}_{\mathbf{T}}(T_{cl}) + \bar{r}_{cl} \left(\cos(\eta), \sin(\eta), 0\right), \tag{36}$$

where $\mathbf{x}_{\mathbf{T}}(T_{cl})$ is given by Equation (30).

To determine the time to classification T_{cl}, it is necessary to replace the interception point $\mathbf{x}_I$ with $\mathbf{x}_{cl}$ in Equations (13) and (17), which once again results in Equations (32) to (34). Since Equation (36) is independent of $\mathbf{x}_{\text{out}}(T_{cl})$, unlike earlier, the analysis in Section 3 still remains valid after this substitution. This is because evaluating Equation (32) using Equation (36) yields an expression for $t_l(T_{cl})$ that is equivalent to Equation (14), that is,

$$t_l(T_{cl}) = \sqrt{(\epsilon T_{cl})^2 + 2\bar{\beta}\epsilon T_{cl} + \bar{\alpha}},$$

where all that has changed are the definitions of α and β:[13]

$$\bar{\alpha} = \left|\mathbf{x}_{\mathbf{T}}^0 + \bar{r}_{cl} \left(\cos(\eta), \sin(\eta), 0\right) - \mathbf{x}_c\right|^2 - 1,$$

$$\bar{\beta} = \mathbf{u}_{\mathbf{T}} \cdot \left(\mathbf{x}_{\mathbf{T}}^0 + \bar{r}_{cl} \left(\cos(\eta), \sin(\eta), 0\right) - \mathbf{x}_c\right).$$

[11] In dimensional units, if $\bar{r}_{cl}/C_{\mathbf{P}} < t_l(T)$ then $T_{cl} = T - \bar{r}_{cl}/C_{\mathbf{P}}$.

[12] This representation of $\mathbf{x}_{cl}$ is mathematically consistent with the technique employed in Section 2.2.

[13] The physical meaning of $\bar{\alpha}$ and $\bar{\beta}$ differs from that of α and β.

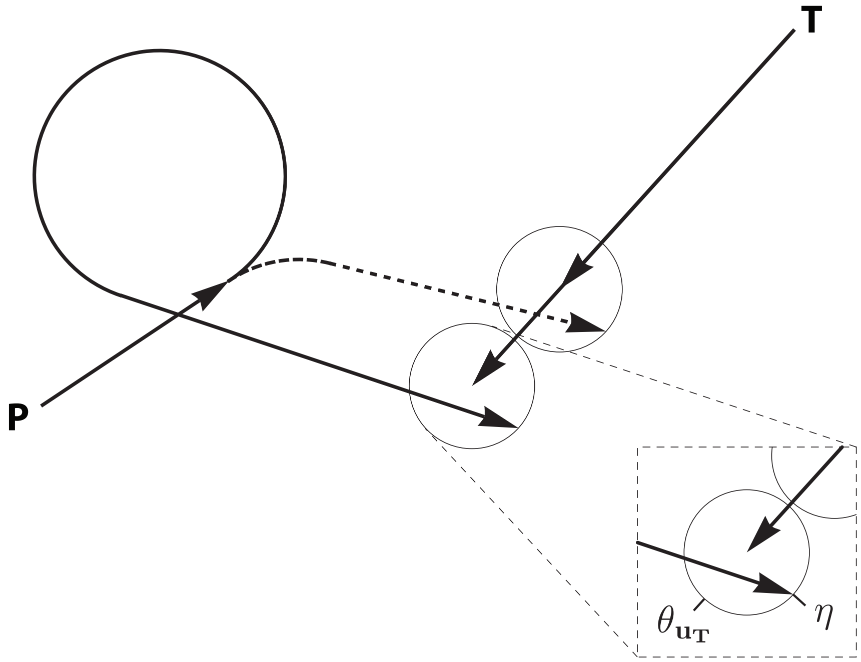


Figure 5: *Two feasible paths to classification resulting from left and right turns. In this figure, the pursuer (P) classifies the target (T) at $\pi/2$ radians on the target's left, that is, $\eta = \theta_{\mathbf{u_T}} + \pi/2$ in Equation (36).*

Hence the *Interception* algorithm can be easily modified to accommodate an angle of approach.

The angle of approach can be chosen relative to the heading of the target, enabling classification to occur on the left or right of the target; see Figure 5. Let $\theta_{\mathbf{u_T}}$ be the heading of the target in polar coordinates centred on the target's position. The definition of the scalar product then gives

$$\theta_{\mathbf{u_T}} = \begin{cases} \arccos(\mathbf{u_T} \cdot \mathbf{i}), & \mathbf{u_T} \cdot \mathbf{j} > 0 \\ 2\pi - \arccos(\mathbf{u_T} \cdot \mathbf{i}), & \mathbf{u_T} \cdot \mathbf{j} \leqslant 0, \end{cases}$$

since $|\mathbf{u_T}| = 1$, where $\mathbf{i} = (1, 0, 0)$ and $\mathbf{j} = (0, 1, 0)$. Now set $\eta = \theta_{\mathbf{u_T}} + \theta$ in Equation (36). If $\theta > 0$ then classification will occur at θ radians to the left of the target, whereas if $\theta < 0$ then classification will occur at θ radians to the right of the target.

6 Conclusion

A pursuer intercepting a target has been considered, where the pursuer and target move at constant speeds in two dimensions and the target's velocity is constant; refer to Figure 1. The pursuer was subjected to Constraints (C1)–(C3); in particular, the pursuer was limited to at most one complete turn. The minimum feasible path to interception for a given turning radius was sought.

A rigorous analysis of Constraints (C1)–(C3) produced the *Interception* algorithm (refer to Figure 3) that can be used to systematically search the feasible region for the minimum time to interception. At the core of *Interception* is a single implicit equation for the minimum time to interception. This equation is valid in an arbitrary Cartesian coordinate system and encompasses both left and right turns. For the purpose of validation, *Interception* has been implemented as a Mathematica package that displays the minimum feasible path to interception, as shown in Figure 4(a)–(f).

Three methods have been proposed for incorporating classification range into the present model: an exact method, a heuristic method, and a method that includes an angle of approach. These methods provide simple models of the pursuer's sensor performance. The *Interception* algorithm can be easily modified to encompass the heuristic and angle of approach methods.

The point-to-point and unconstrained movement of entities in some simulations of military operations is unrealistic. The fidelity of this representation may be insufficient for operations research studies. The *Interception* algorithm enables the effect of turning radius to be incorporated as a constraint into these simulations, improving their fidelity. *Interception* is also straightforward to implement when compared with, for example, a traditional flight dynamics model, and has a broad range of applications in path optimisation problems, the development of computer games and robotics.

To summarise, the main contributions of this paper are

- the reduction of the interception scenario to a single equation;
- the determination of the feasible region as a function of the scenario's inputs, and;
- the *Interception* algorithm, which is a representation of an entity's dynamics in terms of its turning radius.

Acknowledgements

This work was stimulated by discussions with David Marlow, Philip Kilby, Geoff Mercer and Steven Barry, at the 2007 Mathematics-in-Industry Study Group. In particular I thank David Marlow for introducing me to this problem, and for his assistance in deriving Equation (19).

During the preparation of this paper I received invaluable feedback from Russell Connell, Doug Driscoll, Emily Duane, Simon Goss, David Marlow and Michael Papasimeon. This work would not have come to fruition without their input. I also express my thanks to Len Halprin and Josef Zuk for drawing my attention to the literature on classical pursuit curves and differential games, respectively.

References

Atkinson, K. E. (1989) *An introduction to numerical analysis*, 2nd edn, Wiley, New York.

Barton, J. C. & Eliezer, C. J. (2000) On pursuit curves, *J. Austral. Math. Soc. Ser. B* **41**, 358–371.

Boole, G. A. (1859) *Treatise on differential equations*, Macmillan and Co., Cambridge. p. 246.

Colman, W. J. A. (1991) A curve of pursuit, *Bulletin of the Institute of Mathematics and its Applications* **27**(3), 45–47.

Dubins, L. E. (1957) On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents, *American Journal of Mathematics* **79**(3), 497–516.

Eliezer, C. J. & Barton, J. C. (1992) Pursuit curves, *Bulletin of the Institute of Mathematics and its Applications* **28**(11), 182–184.

Eliezer, C. J. & Barton, J. C. (1995) Pursuit curves II, *Bulletin of the Institute of Mathematics and its Applications* **31**(9), 139–141.

Isaacs, R. (1965) *Differential games; a mathematical theory with applications to warfare and pursuit, control and optimization*, Wiley, New York.

Laumond, J., Jacobs, P. E., Taïx, M. & Murray, R. M. (1994) A motion planner for nonholonomic mobile robots, *IEEE Transactions on Robotics and Automation* **10**(5), 577–593.

Marlow, D. O., Kilby, P. & Mercer, G. N. (2007) The travelling salesman problem in maritime surveillance — techniques, algorithms and analysis, *in MODSIM 2007 International Congress on Modelling and Simulation*, Modelling and Simulation Society of Australia and New Zealand, pp. 684–690. ISBN: 978-0-9758400-4-7.

Mercer, G. N., Barry, S. I., Marlow, D. O. & Kilby, P. (2008) Investigating the effect of detection and classification range and aircraft dynamics on a simplified maritime surveillance scenario, *ANZIAM J.* **49**, C475–C492.

Pinter, M. (2001) Toward more realistic pathfinding, Gamasutra.com, `http://www.gamasutra.com/features/20010314/pinter_pfv.htm`, accessed November 2008.

Pinter, M. (2002) *Realistic turning between waypoints*, AI game programming wisdom, Charles River Media, Inc., Hingham, Massachusetts, chapter 4.4, pp. 186–192.

Reeds, J. A. & Shepp, L. A. (1990) Optimal paths for a car that goes both forwards and backwards, *Pacific Journal of Mathematics* **145**(2), 367–393.

Savla, K., Frazzoli, E. & Bullo, F. (2008) Traveling Salesperson Problems for the Dubins Vehicle, *IEEE Transactions on Automatic Control* **53**(6), 1378–1391.

Shima, T. & Shinar, J. (2002) Time-varying linear pursuit-evasion game models with bounded controls, *Journal of Guidance, Control, and Dynamics* **25**(3), 425–432.

Shinar, J., Guelman, M. & Green, A. (1989) An optimal guidance law for a planar pursuit-evasion game of kind, *Computers Math. Applic.* **18**(1–3), 35–44.

Spiegel, M. R. (1974) *Schaum's outline of theory and problems of vector analysis: and an introduction to tensor analysis*, Schaum's outline series, SI (metric) edn, McGraw-Hill, New York.

Weisstein, E. W. (2008) "Pursuit Curve." From MathWorld—A Wolfram Web Resource, `http://mathworld.wolfram.com/PursuitCurve.html`, accessed November 2008.

Appendix A Determining feasible times to interception: the proofs

The primary aim of this appendix is to rigorously determine which times satisfy all of Constraints (C1)–(C3).

A.1 Preliminary results

In Section 2.3 it is claimed that the solutions to

$$T - t_l(T) = 0, \tag{A1}$$

$$T - t_l(T) = \pi, \tag{A2}$$

are given by Equations (22) to (24). These solutions were constructed by solving

$$T^2 = t_l^2(T), \tag{A3}$$

$$(T - \pi)^2 = t_l^2(T). \tag{A4}$$

Solutions of Equations (A1) and (A2) are also solutions of Equations (A3) and (A4), respectively, however the reverse implication is not necessarily true. The main purpose of the following Propositions is to establish when solutions of Equations (A3) and (A4) are also solutions of Equations (A1) and (A2), respectively.

Proposition 1 *Real and nonnegative solutions of Equation (A3) also solve Equation (A1).*

Proof Let T be a real and nonnegative solution of Equation (A3). This implies that $t_l^2(T)$ is also real and nonnegative. Therefore

$$T = |T| = \sqrt{T^2} = \sqrt{t_l^2(T)} = |t_l(T)| = t_l(T).$$

□

Proposition 2 *Let T be a real solution of Equation (A4) such that $T \geqslant \pi$. Then $T - \pi = t_l(T)$.*

Proof Let T be a real solution of Equation (A4) such that $T \geqslant \pi$. This implies that $t_l^2(T)$ is also real and nonnegative. Therefore

$$T - \pi = |T - \pi| = \sqrt{(T - \pi)^2} = \sqrt{t_l^2(T)} = |t_l(T)| = t_l(T).$$

□

A.2 Feasible times to interception when $\alpha > 0$

Recall that the target is initially outside the pursuer's turning-circle if and only if $\alpha > 0$. In Section 2.3 it was shown that Constraint (C1) is satisfied by construction, and there are essentially two cases effecting the feasibility of times to interception with respect to Constraints (C2) and (C3) when $\alpha > 0$: the target will never enter the interior of the pursuer's turning-circle ($\epsilon = 0$ or $\beta \geqslant 0$ or $\beta^2 \leqslant \alpha$), and; the target will enter the interior of the pursuer's turning-circle at some time ($0 < \epsilon < 1$ and $\beta < 0$ and $\beta^2 > \alpha$).

Proposition 3 *Let $0 \leqslant \epsilon < 1$ and $\alpha > 0$. Then T_0 as defined by Equation (22) is a real, strictly positive number and uniquely satisfies $t_c(T_0) = 0$.*

Proof The conditions of the proposition directly give that T_0 is a real number as $(1-\epsilon^2)\alpha > 0$. If $\beta \geqslant 0$ then $T_0 > 0$. If $\beta < 0$ then the inequality $-|X| + \sqrt{X^2 + Y} > 0$ for $Y > 0$ yields $T_0 > 0$. It can be shown by direct substitution that T_0 solves Equation (A3) and hence $t_c(T_0) = 0$, by Proposition 1. A second solution to Equation (A3) does exist, namely

$$\overline{T}_0 = \frac{\epsilon\beta - \sqrt{(\epsilon\beta)^2 + (1-\epsilon^2)\alpha}}{1-\epsilon^2}. \tag{A5}$$

However, by using the aforementioned method, it can be shown that $\overline{T}_0 \leqslant 0$. □

Proposition 4 *Let $\alpha > 0$, $0 < \epsilon < 1$, $\beta < 0$ and $\beta^2 > \alpha$. Then $T_0 \leqslant T_L$.*

Proof Recall that T_0 is a real and strictly positive number, by Proposition 3. Under the conditions of the proposition,

$$\left(1-\epsilon^2\right)\left(\epsilon T_0 + \beta\right) = \beta + \epsilon\sqrt{\epsilon^2(\beta^2 - \alpha) + \alpha} < \beta + |\beta| = 0,$$

and so $T_0 < -\beta/\epsilon$. It follows that $T_0 \leqslant T_L$, by Proposition 3. □

Proposition 5 *Let $\alpha > 0$ and if $\epsilon = 0$ or $\beta \geqslant 0$ or $\beta^2 \leqslant \alpha$, then $t_c(T) = T - t_l(T)$ is a strictly increasing function for $T \geqslant T_0$. If $0 < \epsilon < 1$ and $\beta < 0$ and $\beta^2 > \alpha$, then $t_c(T)$ is strictly increasing for $T_0 \leqslant T \leqslant T_L$.*

Proof Let $\epsilon = 0$ or $\beta \geqslant 0$ or $\beta^2 \leqslant \alpha$. If $\epsilon = 0$ then $t_c(T) = T - \sqrt{\alpha}$ and is strictly increasing. Now let $0 < \epsilon < 1$. If $\beta^2 = \alpha$ then it can be shown that $t_c'(T) = 1 - \operatorname{sgn}(\epsilon T + \beta)\epsilon$ and hence $t_c(T)$ is strictly increasing. Note that

$$t_c'(T) = \frac{t_l(T) - \epsilon(\epsilon T + \beta)}{t_l(T)}. \tag{A6}$$

If $\beta^2 > \alpha$ then $t_c'(T) = 0$ iff

$$t_l(T) = \epsilon(\epsilon T + \beta), \tag{A7}$$

which has the solution

$$\overline{T} = \frac{-(1-\epsilon^2)\beta + \sqrt{(1-\epsilon^2)(\beta^2 - \alpha)}}{\epsilon(1-\epsilon^2)}. \tag{A8}$$

If $\beta^2 < \alpha$ then Equation (A8) implies that $t_c(T)$ has no real stationary points. Let $\beta \geqslant 0$ and $\beta^2 > \alpha$. In this instance it can be shown that $(1-\epsilon^2)(\overline{T}-T_0) < T_R < 0$ and therefore $t_c(T)$ has no stationary points for $T \geqslant T_0$. Proposition 3 yields $t_c'(T_0) > 0$, which concludes the first part of the proof. Now let $0 < \epsilon < 1$ and $\beta < 0$ and $\beta^2 > \alpha$. Since $T_0 \leqslant T_L$ (by Proposition 4) and $T_L < \overline{T}$, $t_c(T)$ has no stationary points for $T_0 \leqslant T \leqslant T_L$. Then, since $t_c'(T_0) > 0$, $t_c(T)$ is strictly increasing for $T_0 \leqslant T \leqslant T_L$. □

Proposition 6 *Let $\alpha > 0$, $0 < \epsilon < 1$, $\beta < 0$, $\beta^2 > \alpha$ and $T_L \leqslant \pi$. Then feasible times to interception T occur in the interval $T_0 \leqslant T \leqslant T_L$.*

Proof Under the present conditions, $t_l(T)$ is real and non-negative for $T_0 \leqslant T \leqslant T_L$ [see Equations (14) and (21)] and therefore Constraint (C2) is satisfied. By Proposition 5, $t_c(T)$ is strictly increasing for $T_0 \leqslant T \leqslant T_L$. Furthermore, $t_c(T_0) = 0$ (by Proposition 3) and $t_c(T_L) = T_L \leqslant \pi$ (by definition) and hence Constraint (C3) is satisfied for $T_0 \leqslant T \leqslant T_L$.□

Proposition 7 *Let $\alpha > 0$ and $\epsilon = 0$ or $\beta \geqslant 0$ or $\beta^2 \leqslant \alpha$. Then T_π is a real number that uniquely satisfies $t_c(T_\pi) = \pi$ and feasible times to interception T occur in the interval $T_0 \leqslant T \leqslant T_\pi$.*

Proof If $\epsilon = 0$ then $T_\pi = \pi + \sqrt{\alpha}$ and is a real number. If $\beta \geqslant 0$ then $t_l^2(T)$ is an increasing function for $T \geqslant 0$ and therefore

$$t_l^2(\pi) + \epsilon^2(\beta^2 - \alpha) \geqslant (\epsilon\beta)^2 + \left(1-\epsilon^2\right)\alpha > 0,$$

implying T_π is real. If $\beta^2 \leqslant \alpha$ then T_π is real since

$$t_l^2(\pi) + \epsilon^2(\beta^2 - \alpha) = (\epsilon\pi + \beta)^2 + \left(1-\epsilon^2\right)\left(\alpha - \beta^2\right) \geqslant 0.$$

Under the present conditions, $t_l(T)$ is real and non-negative for all $T \geqslant 0$ [see Equations (14) and (21)] and therefore Constraint (C2) is satisfied. It remains to show that Constraint (C3) is satisfied. Under the conditions of the proposition, $t_c(T)$ is strictly increasing for $T \geqslant T_0$ (by Proposition 5) and since $t_c(T_0) = 0$ (by Proposition 3), there exists a unique solution $\widetilde{T} > T_0$ to $t_c(\widetilde{T}) = \pi$. Recall that $\widetilde{T}$ also solves Equation (A4), which has two real solutions, T_π and $\overline{T}_\pi$, that are given by Equations (23) and (24), respectively. It follows that $\widetilde{T} = T_\pi$ or $\widetilde{T} = \overline{T}_\pi$. Using Equation (A6) it can be shown that $t_c'(\overline{T}_\pi) \leqslant 0$, however this violates Proposition 5 and hence $\widetilde{T} \neq \overline{T}_\pi$. Therefore $t_c(T_\pi) = t_c(\widetilde{T}) = \pi$. Finally, since $t_c(T)$ is strictly increasing for $T \geqslant T_0$, $T_0 < T_\pi$ and $0 \leqslant t_c(T) \leqslant \pi$ for $T_0 \leqslant T \leqslant T_\pi$. □

Proposition 8 *Let $\alpha > 0$ and $\epsilon = 0$ or $\beta \geqslant 0$ or $\beta^2 \leqslant \alpha$. Then T_π is a real number and there exists a feasible solution T to Equation (18) with $T_0 \leqslant T \leqslant T_\pi$.*

Proof Define the function $F(T)$ to be

$$F(T) = \cos(T - t_l(T)) - (\mathbf{x}_{\text{in}} - \mathbf{x}_c) \cdot (\mathbf{x}_{\text{out}}(T) - \mathbf{x}_c).$$

Note that a zero of $F(T)$ is a solution of Equation (18). Under the conditions of the proposition, $t_l(T)$ is real and continuous, implying $F(T)$ is real and continuous. Observe that $F(T_0) \geqslant 0$ and $F(T_\pi) \leqslant 0$, since by definition

$$-1 \leqslant (\mathbf{x}_{\text{in}} - \mathbf{x}_c) \cdot (\mathbf{x}_{\text{out}}(T) - \mathbf{x}_c) \leqslant 1.$$

It follows from the Intermediate Value Theorem and Propositions 3 and 7, that there exists a feasible solution T to Equation (18) with $T_0 \leqslant T \leqslant T_\pi$. □

Proposition 9 *Let $\alpha > 0$, $0 < \epsilon < 1$, $\beta < 0$, $\beta^2 > \alpha$ and $T_L \geqslant \pi$. Then T_π is a real number that uniquely satisfies $t_c(T_\pi) = \pi$, $T_L \geqslant T_\pi$ and there exists a feasible solution T to Equation (18) with $T_0 \leqslant T \leqslant T_\pi$.*

Proof Rearranging $T_L \geqslant \pi$ leads to $-(\epsilon\pi+\beta) \geqslant \sqrt{\beta^2 - \alpha}$, implying $\epsilon\pi+\beta < 0$. Therefore $(\epsilon\pi + \beta)^2 \geqslant \beta^2 - \alpha > (1 - \epsilon^2)(\beta^2 - \alpha)$, that is, $(\epsilon\pi + \beta)^2 + (1 - \epsilon^2)(\alpha - \beta^2) > 0$ and so T_π is real. Observe that $t_c(T_0) = 0$ (by Proposition 3), $t_c(T_L) = T_L \geqslant \pi$, and t_c is continuous and strictly increasing for $T_0 \leqslant T \leqslant T_L$ (by Proposition 5). It follows that there exists a unique solution to $t_c(\widetilde{T}) = \pi$ with $T_0 < \widetilde{T} \leqslant T_L$. Recall that $\widetilde{T}$ also solves Equation (A4), which has two real solutions, T_π and $\overline{T}_\pi$, that are given by Equations (23) and (24), respectively. It follows that $\widetilde{T} = T_\pi$ or $\widetilde{T} = \overline{T}_\pi$. Using Equation (A6) it can be shown that $t_c'(\overline{T}_\pi) \leqslant 0$, however this violates Proposition 5 and hence $\widetilde{T} \neq \overline{T}_\pi$. Therefore $t_c(T_\pi) = t_c(\widetilde{T}) = \pi$, $T_L \geqslant T_\pi$ and $0 \leqslant t_c(T) \leqslant \pi$ for $T_0 \leqslant T \leqslant T_\pi$. Then by following the reasoning in the proofs of Propositions 7 and 8, it can be shown that there exists a feasible solution T to Equation (18) with $T_0 \leqslant T \leqslant T_\pi$. □

Proposition 10 *Let $\alpha > 0$, $0 < \epsilon < 1$, $\beta < 0$, $\beta^2 > \alpha$ and $T_R \leqslant \pi$. Then T_π is a real number that satisfies $t_c(T_\pi) = \pi$ and feasible times to interception T occur in $T_0 \leqslant T \leqslant T_L$ and $T_R \leqslant T \leqslant T_\pi$.*

Proof Rearranging $T_R \leqslant \pi$ leads to $\epsilon\pi + \beta \geqslant \sqrt{\beta^2 - \alpha}$, implying $\epsilon\pi + \beta > 0$. As in the proof of Proposition 9, it follows that T_π and $\overline{T}_\pi$ are real. Since $T_L < T_R \leqslant \pi$, feasible times to interception occur in $T_0 \leqslant T \leqslant T_L$, by Proposition 6. Recall that $\overline{T}$ is the unique stationary point of $t_c(T)$ and $\overline{T} > T_R$ [see Equations (21) and (A8)]. Furthermore

$$t_c''(T) = \frac{\epsilon^2(\beta^2 - \alpha)}{t_l^3(T)}, \tag{A9}$$

and hence $t_c(T)$ has a local minimum at $\overline{T}$. Observe that

$$t_c(\overline{T}) = \frac{-\beta + \sqrt{(1 - \epsilon^2)(\beta^2 - \alpha)}}{\epsilon}, \tag{A10}$$

and $0 < t_c(\overline{T}) < T_R \leqslant \pi$, implying that Constraint (C3) is satisfied for $T_R \leqslant T \leqslant \overline{T}$. Since $0 < t_c(\overline{T}) < \pi$ and t_c is continuous and strictly increasing for $T > \overline{T}$, there exists a unique solution $\widetilde{T} > \overline{T}$ to $t_c(\widetilde{T}) = \pi$. By following the argument in the proof of Proposition 9, it can be shown that $\widetilde{T} = T_\pi$. Furthermore, $T_\pi > \overline{T}$ and Constraint (C3) is satisfied for $\overline{T} < T \leqslant T_\pi$. □

Proposition 11 *Let T_π be a complex number or $\epsilon\pi + \beta \leqslant 0$. Furthermore, let $\alpha > 0$, $0 < \epsilon < 1$, $\beta < 0$ and $\beta^2 > \alpha$. Then $T_L < \pi < T_R$ and feasible times to interception T only occur in $T_0 \leqslant T \leqslant T_L$.*

Proof By Propositions 9 and 10, if $T_L \geqslant \pi$ or $T_R \leqslant \pi$ then T_π is real, which under the present conditions implies that $T_L < \pi < T_R$. It follows from Proposition 6 that feasible times to interception occur in $T_0 \leqslant T \leqslant T_L$. It remains to prove there does not exist any feasible times to interception for $T > T_L$. Constraint (C2) is not satisfied for times $T_L < T < T_R$. Now let $T \geqslant T_R$. If T_π is complex then Equation (23) yields $|\epsilon\pi + \beta| < \sqrt{(1-\epsilon^2)(\beta^2-\alpha)}$. It follows that $-(\epsilon\pi+\beta) + \sqrt{(1-\epsilon^2)(\beta^2-\alpha)} > 0$, as $x \leqslant |x|$ for any real number x. Then from Equation (A10),

$$\epsilon(t_c(\overline{T}) - \pi) = -(\epsilon\pi + \beta) + \sqrt{(1-\epsilon^2)(\beta^2-\alpha)} > 0, \tag{A11}$$

and so $t_c(\overline{T}) > \pi$. Recall that $t_c(\overline{T})$ is the global minimum of $t_c(T)$ for $T > T_R$ and $t_c(T_R) = T_R > \pi$, therefore Constraint (C3) is violated for $T \geqslant T_R$. If $\epsilon\pi + \beta \leqslant 0$ then Equation (A11) implies that Constraint (C3) is violated for $T \geqslant T_R$. □

Proposition 12 *Let T_π be a real number, $\epsilon\pi + \beta > 0$, $\alpha > 0$, $0 < \epsilon < 1$, $\beta < 0$, $\beta^2 > \alpha$ and $T_L < \pi < T_R$. Then $t_c(T_\pi) = t_c(\overline{T}_\pi) = \pi$ and feasible times to interception T occur in $T_0 \leqslant T \leqslant T_L$ and $\overline{T}_\pi \leqslant T \leqslant T_\pi$.*

Proof Since $T_L < \pi$, feasible times to interception occur in $T_0 \leqslant T \leqslant T_L$, by Proposition 6. Since T_π is real, Equation (23) yields $|\epsilon\pi + \beta| \geqslant \sqrt{(1-\epsilon^2)(\beta^2-\alpha)}$. As $\epsilon\pi + \beta > 0$, it follows that

$$-(\epsilon\pi + \beta) + \sqrt{(1-\epsilon^2)(\beta^2-\alpha)} \leqslant 0,$$

and so $t_c(\overline{T}) \leqslant \pi$ [see Equation (A11)]. Recall that $t_c(\overline{T})$ is the global minimum of $t_c(T)$ for $T > T_R$. Furthermore, $t_c(T_R) = T_R > \pi$ and $T_R < \overline{T}$. Hence there exist real solutions $\widetilde{T}_1$, $\widetilde{T}_2$ to $t_c(\widetilde{T}_i) = \pi$ (for $i = 1, 2$) such that $T_R < \widetilde{T}_1 \leqslant \overline{T}$ and $\overline{T} \leqslant \widetilde{T}_2$. Under the present conditions, Equation (A4) has exactly two real solutions T_π, $\overline{T}_\pi$, and $\widetilde{T}_1$, $\widetilde{T}_2$ also solve Equation (A4). Therefore $T_\pi = \widetilde{T}_1$ and $\overline{T}_\pi = \widetilde{T}_2$, and feasible times to interception occur in $\overline{T}_\pi \leqslant T \leqslant T_\pi$. □

A.3 Feasible times to interception when $\alpha \leqslant 0$

Recall that the target is initially inside (or on the boundary of) the pursuer's turning-circle if and only if $\alpha \leqslant 0$. Then Constraint (C2) is only satisfied for times $T \geqslant T_R$, because the target does not leave the pursuer's turning-circle until $T = T_R$.

Proposition 13 *Let $\alpha \leqslant 0$ and $0 < \epsilon < 1$. Then $T_L \leqslant 0$ and $T_R \geqslant 0$.*

Proof The proof follows from the definitions of T_L and T_R and the inequality $|\beta| \leqslant \sqrt{\beta^2 - \alpha}$. □

Proposition 14 *Let $\alpha \leqslant 0$ and $0 < \epsilon < 1$. Furthermore, let T_π be a complex number or $\epsilon\pi + \beta \leqslant 0$. Then feasible times to interception do not exist.*

Proof Since $\alpha \leqslant 0$ and $0 < \epsilon < 1$, feasible times to interception cannot exist in the domain $T < T_R$ (see Proposition 13). By following the proof of Proposition 11, it can be seen that feasible times to interception do not exist for $T \geqslant T_R$. □

Proposition 15 *Let $\alpha \leqslant 0$, $0 < \epsilon < 1$, T_0 be a real number and $\beta \geqslant 0$. Then $T_0 \geqslant 0$ and satisfies $t_c(T_0) = 0$, and $\min_{T>T_R} t_c(T) \leqslant 0$.*

Proof The conditions of the proposition and the definition of T_0 give $T_0 \geqslant 0$ and $t_c(T_0) = 0$ (by Proposition 1). Since T_0 is a real number, $\epsilon^2(\beta^2 - \alpha) + \alpha \geqslant 0$. This implies

$$-|\beta| + \sqrt{(1-\epsilon^2)(\beta^2 - \alpha)} \leqslant 0.$$

Recall that $t_c(T)$ has a unique minimum at $\overline{T} > T_R$; see Equation (A8). The above inequality in conjunction with the present conditions and Equation (A10) yield $\min_{T>T_R} t_c(T) = t_c(\overline{T}) \leqslant 0$. □

Proposition 16 *Let $\alpha \leqslant 0$ and $0 < \epsilon < 1$. Furthermore, let T_0 be a complex number or $\beta < 0$. Then $\min_{T>T_R} t_c(T) > 0$.*

Proof Let $\beta < 0$. Then Equation (A10) gives $\min_{T>T_R} t_c(T) = t_c(\overline{T}) > 0$. Now let T_0 be a complex number. Then following the proof of Proposition 15 leads to

$$-\beta + \sqrt{(1-\epsilon^2)(\beta^2 - \alpha)} \geqslant -|\beta| + \sqrt{(1-\epsilon^2)(\beta^2 - \alpha)} > 0,$$

and therefore $\min_{T>T_R} t_c(T) = t_c(\overline{T}) > 0$, by Equation (A10). □

Proposition 17 *Let $\alpha \leqslant 0$, $0 < \epsilon < 1$, $T_R \leqslant \pi$, T_0 and T_π be real numbers and $\beta \geqslant 0$. Then $t_c(\overline{T}_0) = t_c(T_0) = 0$, $t_c(T_\pi) = \pi$, feasible times to interception T occur in $T_R \leqslant T \leqslant \overline{T}_0$, and there exists a feasible solution to Equation (18) with $T_0 \leqslant T \leqslant T_\pi$.*

Proof Recall that for $T > T_R$, t_c is real, continuous and has a unique minimum of $t_c(\overline{T})$. In addition, $t_c(T_R) = T_R \geqslant 0$ and $t_c(\overline{T}) \leqslant 0$, by Propositions 13 and 15. Hence there exist real solutions $\widetilde{T}_1$, $\widetilde{T}_2$ to $t_c(\widetilde{T}_i) = 0$ (for $i = 1, 2$) such that $T_R \leqslant \widetilde{T}_1 \leqslant \overline{T}$ and $\overline{T} \leqslant \widetilde{T}_2$. Under the present conditions, Equation (A3) has exactly two real nonnegative solutions T_0 and $\overline{T}_0$ [see Equation (A5)], and it follows that $\overline{T}_0 = \widetilde{T}_1$ and $T_0 = \widetilde{T}_2$, by Proposition 1. Likewise, it can be shown that $t_c(T_\pi) = \pi$ and $T_0 < T_\pi$. It can be shown using Equation (A6) that t_c is strictly decreasing for $T_R \leqslant T \leqslant \overline{T}_0$ and strictly increasing for $T_0 \leqslant T \leqslant T_\pi$. Then, since $0 \leqslant t_c(T_R) \leqslant \pi$, feasible times to interception occur in $T_R \leqslant T \leqslant \overline{T}_0$ and $T_0 \leqslant T \leqslant T_\pi$. Observe that $t_c(T) < 0$ for $\overline{T}_0 < T < T_0$ and therefore Constraint (C3) is violated in this interval. The existence of a solution to Equation (18) in $T_0 \leqslant T \leqslant T_\pi$ can be seen by following the proof of Proposition 8. □

Proposition 18 *Let $\alpha \leqslant 0$, $0 < \epsilon < 1$, $T_R > \pi$, T_0 and T_π be real numbers and $\beta \geqslant 0$. Then $t_c(\overline{T}_0) = t_c(T_0) = 0$, $t_c(\overline{T}_\pi) = t_c(T_\pi) = \pi$, feasible times to interception T occur in $\overline{T}_\pi \leqslant T \leqslant \overline{T}_0$, and there exists a feasible solution to Equation (18) with $T_0 \leqslant T \leqslant T_\pi$.*

Proof The proof is the same as that of Proposition 17, with the following addition. Since $t_c(T_R) > \pi$ and $t_c(\overline{T}_0) = 0$ (refer to the proof of Proposition 17), it follows that $t_c(\overline{T}_\pi) = \pi$ with $T_R < \overline{T}_\pi < \overline{T}_0$. Furthermore, Constraint (C3) is violated for $T_R \leqslant T < \overline{T}_\pi$ and feasible times to interception occur in $\overline{T}_\pi \leqslant T \leqslant \overline{T}_0$. □

Proposition 19 *Let $\alpha \leqslant 0$, $0 < \epsilon < 1$, $T_R \leqslant \pi$, T_π be a real number and $\epsilon\pi + \beta > 0$. Furthermore, let T_0 be a complex number or $\beta < 0$. Then $t_c(T_\pi) = \pi$ and feasible times to interception T occur in $T_R \leqslant T \leqslant T_\pi$.*

Proof Recall that feasible times to interception do not exist for $T < T_R$, and observe that $0 < t_c(\overline{T}) < T_R \leqslant \pi$ (by Proposition 16). The proof can be completed by following the proof of Proposition 10. □

Proposition 20 *Let $\alpha \leqslant 0$, $0 < \epsilon < 1$, $T_R > \pi$, T_π be a real number and $\epsilon\pi + \beta > 0$. Furthermore, let T_0 be a complex number or $\beta < 0$. Then $t_c(\overline{T}_\pi) = t_c(T_\pi) = \pi$ and feasible times to interception T occur in $\overline{T}_\pi \leqslant T \leqslant T_\pi$.*

Proof Recall that feasible times to interception do not exist for $T < T_R$, and observe that $t_c(\overline{T}) > 0$ (by Proposition 16). The proof can be completed by following the proof of Proposition 12. □

Appendix B An explanatory flow chart of the *Interception* algorithm

Below are descriptions of the errors, warnings and parameters that appear in Figure B1(a)–(b):

Error 1 is returned if the target's speed is greater than or equal to the speed of the pursuer.

Error 2 is returned if the root-finding method used to solve Equation (18) fails to converge, when a solution is known to exist; see Propositions 8 and 9 from Appendix A.2.

Warning 1 is returned if the root-finding method used to solve Equation (18) fails to converge. In this instance, the pursuer is unable to intercept the target before it enters the turning-circle (if $\alpha > 0$), and the target is moving too slowly to leave the turning-circle before the pursuer has performed a complete turn; see Case 3 from Section 3.

Warning 2 is returned if the root-finding method used to solve Equation (18) fails to converge. In this instance, the pursuer is unable to intercept the target before it enters the turning-circle (if $\alpha > 0$), and the target is moving too slowly to leave the turning-circle in time for the pursuer to attempt an intercept; see Cases 1 and 2 from Section 3.

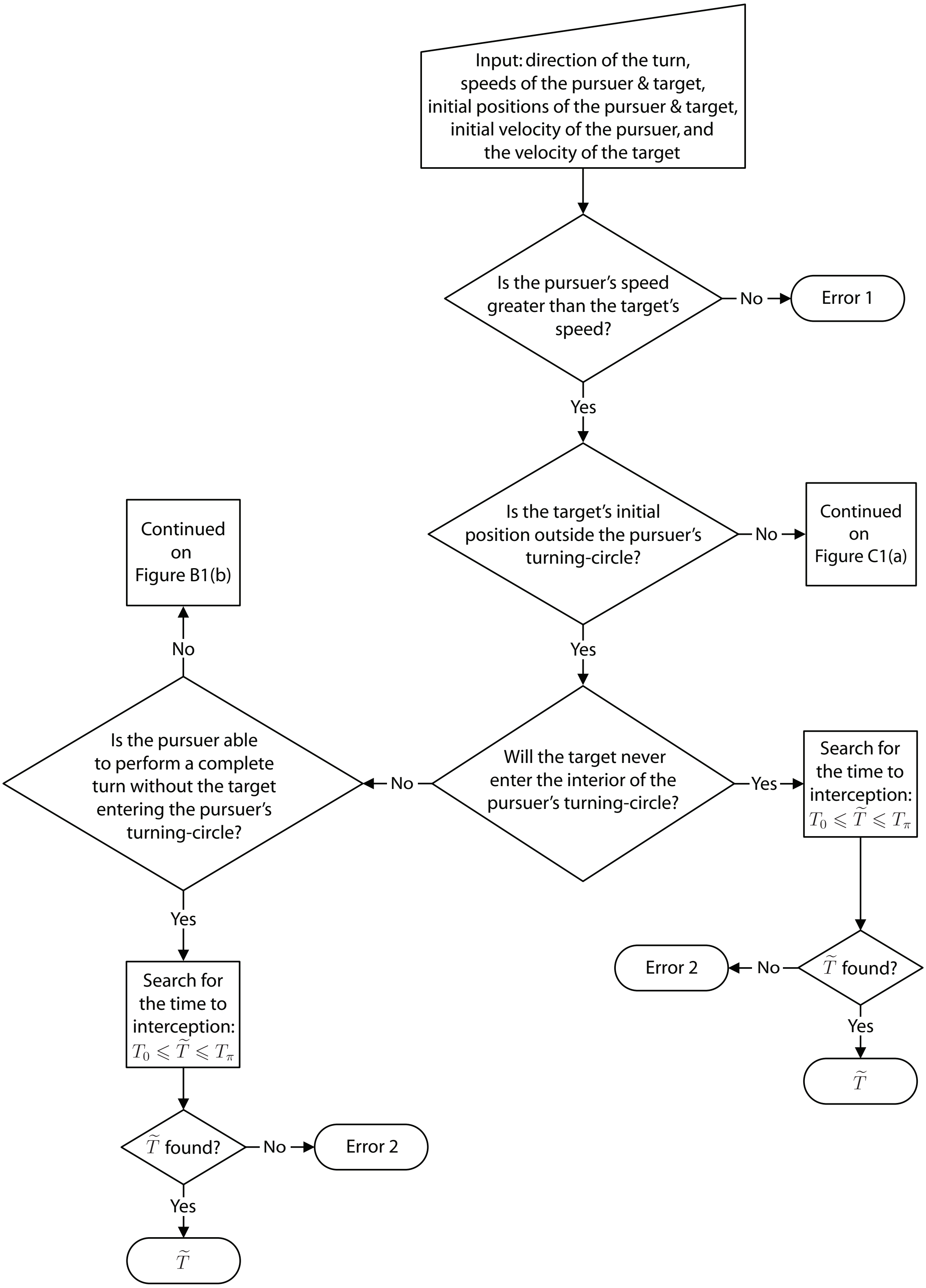
T_0 is the time to interception if the pursuer does not turn [see Equation (22)].

T_L is the time for the target to enter the pursuer's turning-circle [see Equation (21)].

T_R is the time for the target to exit the pursuer's turning-circle [see Equation (21)].

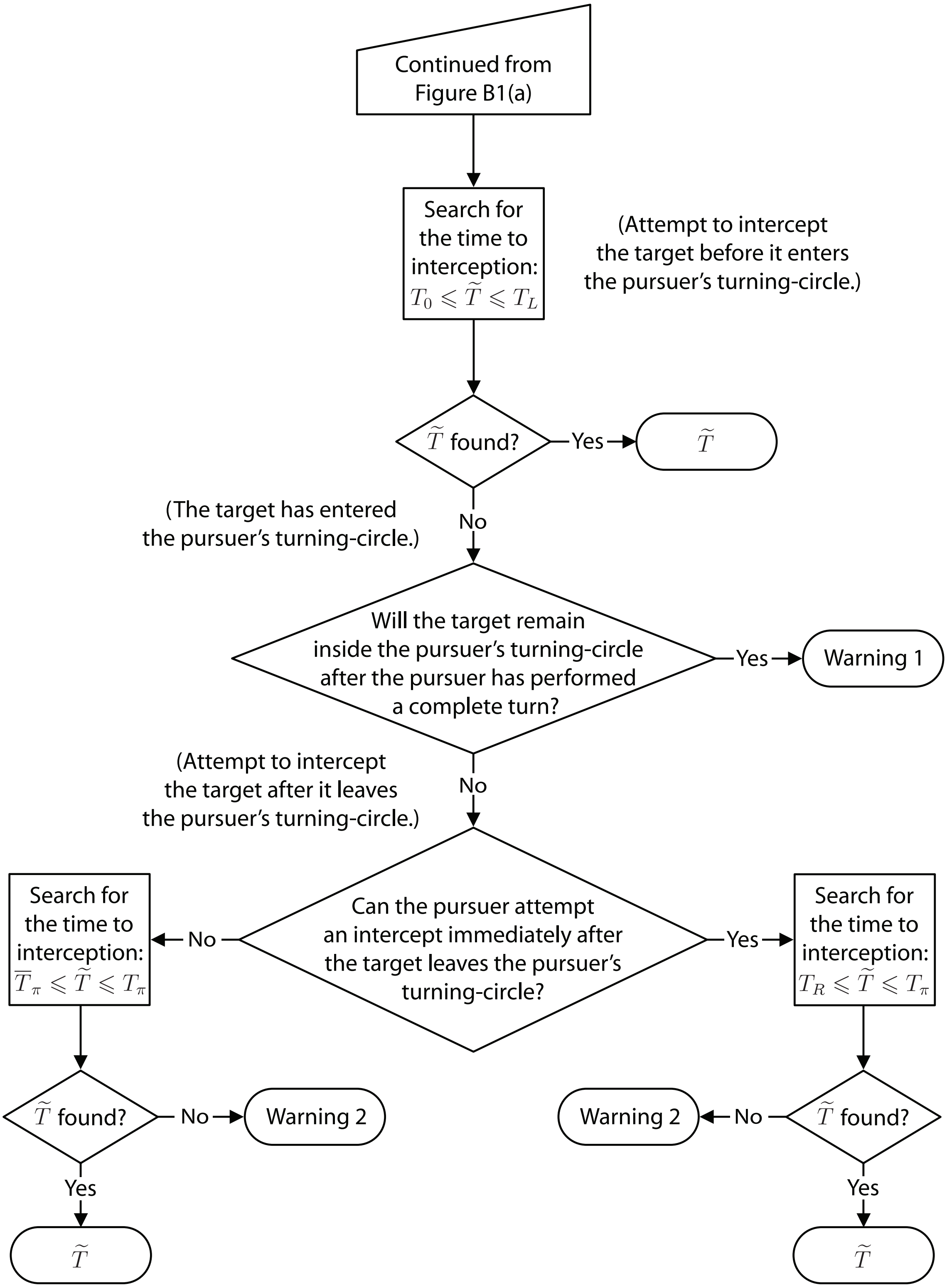
T_π is the time to interception if the pursuer takes π units of time to turn [see Equation (23)].

$\overline{T}_\pi$ is the time to interception if the pursuer takes π units of time to turn [see Equation (24)].


Input: direction of the turn, speeds of the pursuer & target, initial positions of the pursuer & target, initial velocity of the pursuer, and the velocity of the target
Is the pursuer's speed greater than the target's speed?
No
Error 1
Yes
Is the target's initial position outside the pursuer's turning-circle?
No
Continued on Figure C1(a)
Yes
Will the target never enter the interior of the pursuer's turning-circle?
Yes
Search for the time to interception: $T_0 \leqslant \widetilde{T} \leqslant T_\pi$
$\widetilde{T}$ found?
No
Error 2
Yes
$\widetilde{T}$
No
Is the pursuer able to perform a complete turn without the target entering the pursuer's turning-circle?
No
Continued on Figure B1(b)
Yes
Search for the time to interception: $T_0 \leqslant \widetilde{T} \leqslant T_\pi$
$\widetilde{T}$ found?
No
Error 2
Yes
$\widetilde{T}$

(a) The cases where the pursuer is able to perform a complete turn without the target entering the pursuer's turning-circle. The other cases are presented in Figure B1(b).

Figure B1: *The Interception algorithm (continued on Figure B1(b)) for determining feasible times to interception $\widetilde{T}$, described using words rather than symbols; refer to Figure 3 for the symbolic version. Descriptions of the errors, warnings and parameters can be found on page 35.*


Continued from Figure B1(a)
Search for the time to interception: $T_0 \leqslant \widetilde{T} \leqslant T_L$
(Attempt to intercept the target before it enters the pursuer's turning-circle.)
$\widetilde{T}$ found?
Yes
$\widetilde{T}$
No
(The target has entered the pursuer's turning-circle.)
Will the target remain inside the pursuer's turning-circle after the pursuer has performed a complete turn?
Yes
Warning 1
No
(Attempt to intercept the target after it leaves the pursuer's turning-circle.)
Can the pursuer attempt an intercept immediately after the target leaves the pursuer's turning-circle?
No
Yes
Search for the time to interception: $\overline{T}_\pi \leqslant \widetilde{T} \leqslant T_\pi$
Search for the time to interception: $T_R \leqslant \widetilde{T} \leqslant T_\pi$
$\widetilde{T}$ found?
No
Warning 2
Yes
$\widetilde{T}$
Warning 2
No
$\widetilde{T}$ found?
Yes
$\widetilde{T}$

(b) The cases where the target will enter the pursuer's turning-circle before the pursuer completes a turn. The other cases are presented in Figure B1(a).

Figure B1: *The Interception algorithm (continued from Figure B1(a)) for determining feasible times to interception $\widetilde{T}$, described using words rather than symbols; refer to Figure 3 for the symbolic version. Descriptions of the errors, warnings and parameters can be found on page 35.*

Appendix C The *Interception* algorithm when $\alpha \leqslant 0$

The *Interception* algorithm for the case when the target is initially inside (or on the boundary of) the pursuer's turning-circle ($\alpha \leqslant 0$) is presented in Figure C1(a)–(b). Refer to Appendix A.3 for the derivation of *Interception* for the $\alpha \leqslant 0$ case.

Below is a brief description of the algorithm for this case:

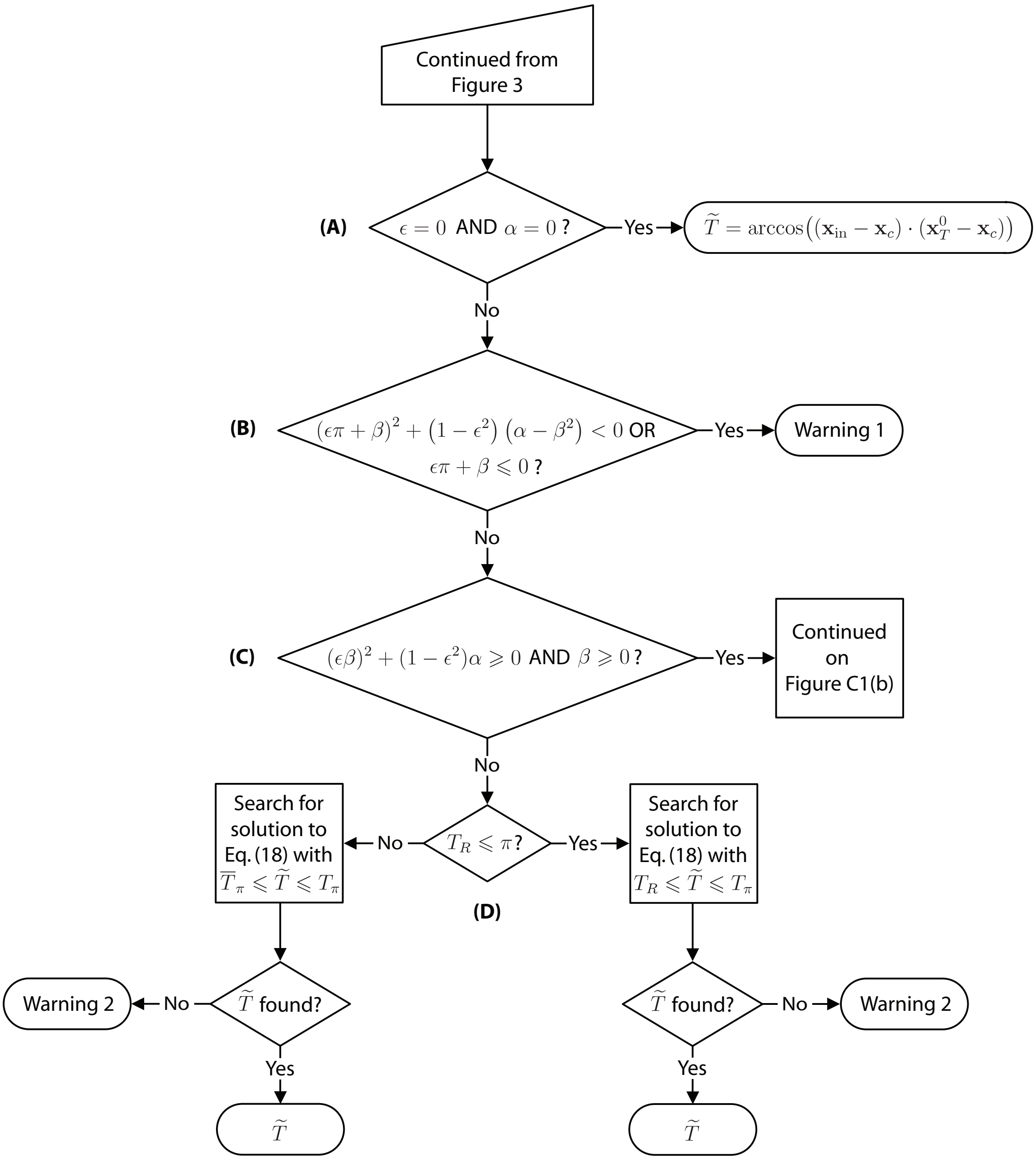
(A): Is the target stationary and initially on the boundary of the pursuer's turning-circle?

(B): Will the target remain inside the pursuer's turning-circle after the pursuer has performed a complete turn?

(C): Is the target near the boundary of the pursuer's turning-circle and heading out of the turning-circle?

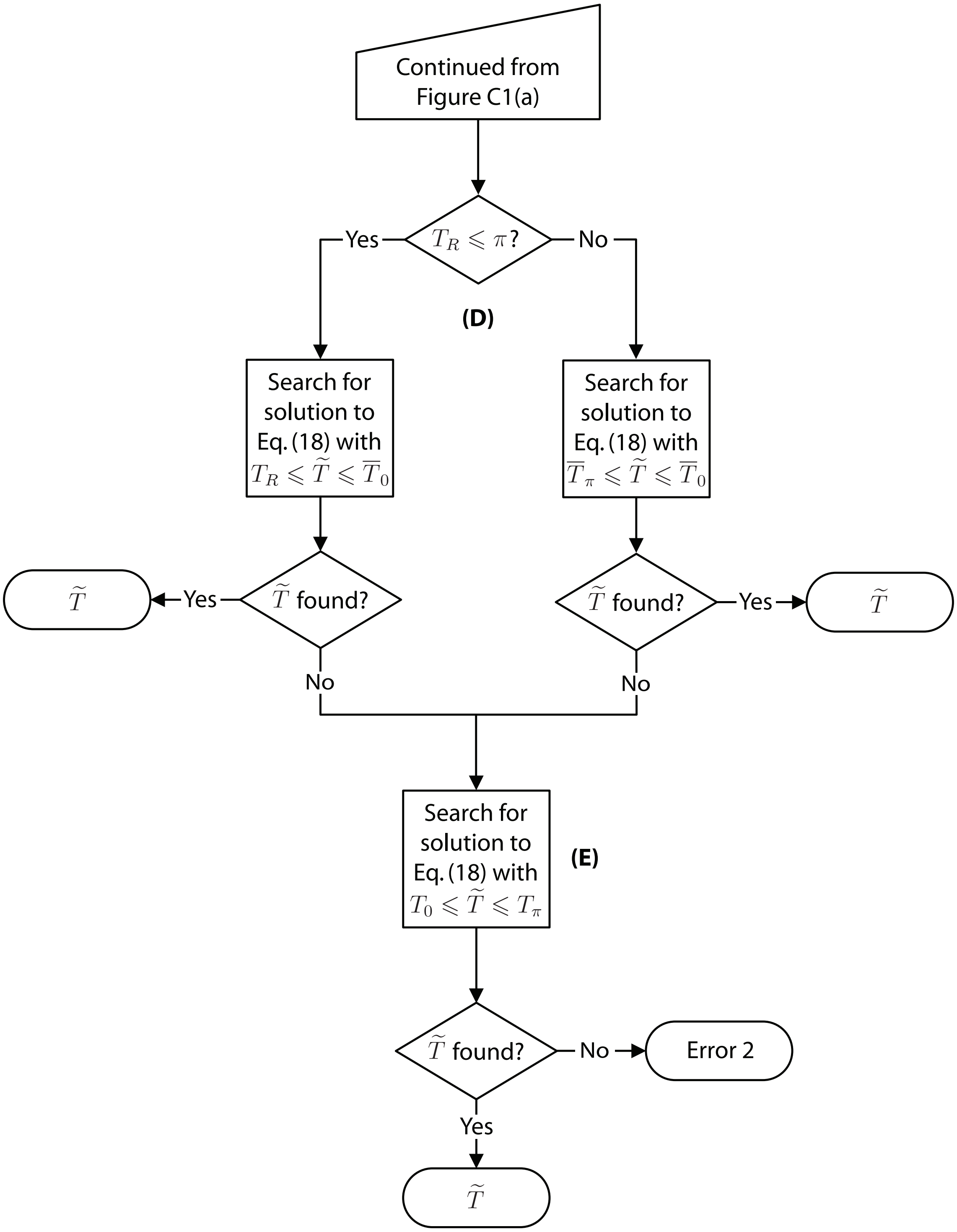
(D): Can the pursuer feasibly attempt an intercept immediately after the target leaves the pursuer's turning-circle?

(E): In this case, the target is near the boundary of the pursuer's turning-circle and heading out of the turning-circle. If the pursuer has not intercepted the target immediately after the target leaves the turning-circle, then the target will be far enough away from the turning-circle for the pursuer to feasibly perform a complete turn before attempting an intercept.


Continued from Figure 3
(A)
$\epsilon = 0$ AND $\alpha = 0$?
Yes
$\widetilde{T} = \arccos\big((\mathbf{x}_{\text{in}} - \mathbf{x}_c) \cdot (\mathbf{x}_T^0 - \mathbf{x}_c)\big)$
No
(B)
$(\epsilon\pi + \beta)^2 + \left(1 - \epsilon^2\right)\left(\alpha - \beta^2\right) < 0$ OR $\epsilon\pi + \beta \leqslant 0$?
Yes
Warning 1
No
(C)
$(\epsilon\beta)^2 + (1 - \epsilon^2)\alpha \geqslant 0$ AND $\beta \geqslant 0$?
Yes
Continued on Figure C1(b)
No
$T_R \leqslant \pi$?
(D)
No
Search for solution to Eq. (18) with $\overline{T}_\pi \leqslant \widetilde{T} \leqslant T_\pi$
Yes
Search for solution to Eq. (18) with $T_R \leqslant \widetilde{T} \leqslant T_\pi$
$\widetilde{T}$ found?
No
Warning 2
Yes
$\widetilde{T}$
$\widetilde{T}$ found?
No
Warning 2
Yes
$\widetilde{T}$

(a) The cases where the target is stationary and on the boundary of the pursuer's turning-circle, or where the target is not near the boundary of the turning-circle or heading into the turning-circle. The other cases are presented in Figure C1(b).

Figure C1: *The continuation of the* Interception *algorithm for determining feasible times to interception $\widetilde{T}$, where the initial position of the target is inside (or on the boundary of) the turning-circle of the pursuer. The Errors and Warnings are described in Section 4.1.3, and refer to the Notation section on page xi for descriptions of the parameters.*


Continued from Figure C1(a)
$T_R \leqslant \pi$?
Yes
No
(D)
Search for solution to Eq. (18) with $T_R \leqslant \widetilde{T} \leqslant \overline{T}_0$
Search for solution to Eq. (18) with $\overline{T}_\pi \leqslant \widetilde{T} \leqslant \overline{T}_0$
$\widetilde{T}$ found?
Yes
$\widetilde{T}$
$\widetilde{T}$ found?
Yes
$\widetilde{T}$
No
No
Search for solution to Eq. (18) with $T_0 \leqslant \widetilde{T} \leqslant T_\pi$
(E)
$\widetilde{T}$ found?
No
Error 2
Yes
$\widetilde{T}$

(b) The case where the target is near the boundary of the pursuer's turning-circle and heading out of the turning-circle. The other cases are presented in Figure C1(a).

Figure C1: *The continuation of the* Interception *algorithm for determining feasible times to interception $\widetilde{T}$, where the initial position of the target is inside (or on the boundary of) the turning-circle of the pursuer. The Errors and Warnings are described in Section 4.1.3, and refer to the Notation section on page xi for descriptions of the parameters.*

Appendix D The Mathematica package: TurningCircle.m

```
BeginPackage["TurningCircle‘"]

TurningCircle::usage="TurningCircle[eps,purinit,tarinit,purvel,tarvel]

The dimensionless parameter eps=ct/cp:
- cp: pursuer’s speed
- ct: target’s speed.

It is assumed that 0 <= eps < 1.

TurningCircle also requires the following inputs to be defined as
2-vectors:

- purinit: pursuer’s position at t=0
- tarinit: target’s position at t=0
- purvel: pursuer’s velocity at t=0 (|purvel|==1)
- tarvel: target’s velocity (|tarvel|==1).

All variables have been scaled:
x by r, t by r/cp, purvel by cp, and tarvel by ct; here r=radius
of the pursuer’s turning-circle.

An optional argument BlackAndDashed->True will display the paths
as black and dashed curves."

TurningCircle::err1="eps must satisfy 0 <= eps < 1"

TurningCircle::err2="|purvel| and |tarvel| must be 1"

TurningCircle::err4="|purinit-tarinit| must be > 0"

TurningCircle::err5="Unable to compute intercept times"

TurningCircle::err7="FindRoot failed to converge"

TurningCircle::warn1="Unable to perform turn"

TurningCircle::warn2="Unable to perform turn"

Options[TurningCircle]={BlackAndDashed->False}

Begin["TurningCircle‘Private‘"]
```

```
i={1,0,0}; j={0,1,0}; k={0,0,1};

(*Centre of the two turning-circles, n=0,1*)
xc[n_]:=xin+(-1)^(n+1)*Cross[up,k]

(*Pursuer's intercept points, where T=tC+tL is the total intercept time,
eps=cs/ca*)
xI[T_,eps_]:=xs+eps*T*us

(*Time from turning-circle exit point to intercept point, tL*)
A[n_]:=If[Abs[Norm[xs-xc[n]]-1]>NumTol,Dot[xs-xc[n],xs-xc[n]]-1,0]
B[n_]:=Dot[us,xs-xc[n]]
tL[T_,eps_,n_]:=Sqrt[(eps*T)^2+2*eps*B[n]*T+A[n]]

(*Pursuer's turning-circle exit points*)
xout[T_,eps_,n_]:=
(xI[T,eps]+tL[T,eps,n]^2*xc[n]+(-1)^n*tL[T,eps,n]*Cross[xI[T,eps]-xc[n],k])/
(1+tL[T,eps,n]^2)

(*Value of T corresponding to tC=0*)
TZero[eps_,n_]:=(eps*B[n]+Sqrt[(eps*B[n])^2+(1-eps^2)*A[n]])/(1-eps^2)
TZeroBar[eps_,n_]:=(eps*B[n]-Sqrt[(eps*B[n])^2+(1-eps^2)*A[n]])/(1-eps^2)

(*Value of T corresponding to tC=Pi*)
TPi[m_,eps_,n_]:=
(eps*B[n]+Pi+(-1)^(m+1)*Sqrt[(eps*Pi+B[n])^2+(1-eps^2)*(A[n]-B[n]^2)])/
(1-eps^2)

(*The left and right endpoints of the solution's domain when B[n] < 0 AND
B[n]^2 >= A[n] AND 0 < eps < 1*)
TLeftRight[m_,eps_,n_]:=(-B[n]+(-1)^(m+1)*Sqrt[B[n]^2-A[n]])/eps

(*Pursuer's interception times: T=tC+tL*)
Interception[eps_,n_]:=(

(*Check Assumptions*)
Which[
  eps<0||eps>=1,
    Message[TurningCircle::err1]; Abort[],
  Norm[up]!=1||Norm[us]!=1,
    Message[TurningCircle::err2]; Abort[],
  Norm[xin-xs]<NumTol,
    Message[TurningCircle::err4]; Abort[]
];

If[A[n]>0,
  AlphaPosInterception[eps,n],
  AlphaNegInterception[eps,n]
]

)

(*If Target starts outside the turning-circle*)
AlphaPosInterception[eps_,n_]:=Module[

(*Local Variables*)
{T,TLeft,TRight},

If[0<eps<1&&B[n]<0&&B[n]^2>A[n],

(*The harder case*)

  If[TLeftRight[0,eps,n]<Pi,
    TRight=TLeftRight[0,eps,n],
    TRight=TPi[1,eps,n]
  ];

  (*Look for solution TZero <= T <= T- OR TZero <= T <= TPi+*)
  T=SearchForT[TZero[eps,n],TRight,eps,n];

  If[NumberQ[T],
    Return[T],
    If[TLeftRight[0,eps,n]>=Pi,
      Return[Message[TurningCircle::err7]]
    ];
  ];

  (*If the Target will not leave the turning-circle in time ... *)
  If[(eps*Pi+B[n])^2+(1-eps^2)*(A[n]-B[n]^2)<0||eps*Pi+B[n]<=0,
    Return[Message[TurningCircle::warn1]]
  ];

  (*If a solution has not been found AND an Abort[] has not occurred,
  look for solution T+ <= T <= TPi+ OR TPi- <= T <= TPi+*)
  If[TLeftRight[1,eps,n]>Pi,
    TLeft=TPi[0,eps,n],
    TLeft=TLeftRight[1,eps,n]
  ];

  T=SearchForT[TLeft,TPi[1,eps,n],eps,n];

  If[NumberQ[T],
    Return[T],
    Message[TurningCircle::warn2]
  ];

,
  (*The easy case*)

  (*Look for solution TZero <= T <= TPi++*)
  T=SearchForT[TZero[eps,n],TPi[1,eps,n],eps,n];

  If[NumberQ[T],
    Return[T],
    Return[Message[TurningCircle::err7]]
  ];

]

(*End AlphaPosInterception*)
]

(*If Target starts inside the turning-circle*)
AlphaNegInterception[eps_,n_]:=Module[

(*Local Variables*)
{T,TLeft},

(*If Target is stationary and on the rim of the turning-circle*)
If[eps==0&&A[n]==0,
  Return[ArcCos[Dot[xin-xc[n],xs-xc[n]]]]
];

(*If Target will not leave turning-circle*)
If[(eps*Pi+B[n])^2+(1-eps^2)*(A[n]-B[n]^2)<0||eps*Pi+B[n]<=0,
  Return[Message[TurningCircle::warn1]]
];

(*If Target is not near the rim or heading into the turning-circle*)
If[(eps*B[n])^2+(1-eps^2)*A[n]<0||B[n]<0,

  If[TLeftRight[1,eps,n]>Pi,
    TLeft=TPi[0,eps,n],
    TLeft=TLeftRight[1,eps,n]
  ];

  T=SearchForT[TLeft,TPi[1,eps,n],eps,n];

  If[NumberQ[T],
    Return[T],
    Return[Message[TurningCircle::warn2]]
  ];

,
  (*Else if Target is near the rim and heading out of the turning-circle*)
  If[TLeftRight[1,eps,n]>Pi,
    TLeft=TPi[0,eps,n],
    TLeft=TLeftRight[1,eps,n]
  ];

  T=SearchForT[TLeft,TZeroBar[eps,n],eps,n];

  If[NumberQ[T],
    Return[T]
  ];

  T=SearchForT[TZero[eps,n],TPi[1,eps,n],eps,n];

  If[NumberQ[T],
    Return[T],
    Return[Message[TurningCircle::err7]]
  ];

]

(*End AlphaNegInterception*)
]

(*Finding the solution*)
SearchForT[xL_,xR_,eps_,n_]:=Module[

(*Local Variables*)
{FracInit=70000,FracPer=0.3,Frac,FR,T},

If[xR<NumTol,Return[]];

Frac=FracInit;
While[!VectorQ[FR]&&Frac>1,
  FR=CheckAbort[
  FindRoot[Cos[T-tL[T,eps,n]]==Dot[xin-xc[n],xout[T,eps,n]-xc[n]],
  {T,(xR+(Frac-1)*xL)/Frac},
  EvaluationMonitor:>If[Im[T]!=0||T<xL||T>xR,Abort[]]],
  Null];
  Frac=FracPer*Frac;
];

(*If unsuccessful, try again from the right*)
If[!VectorQ[FR],
  Frac=FracInit;
  While[!VectorQ[FR]&&Frac>1,
    FR=CheckAbort[
    FindRoot[Cos[T-tL[T,eps,n]]==Dot[xin-xc[n],xout[T,eps,n]-xc[n]],
    {T,(xL+(Frac-1)*xR)/Frac},
    EvaluationMonitor:>If[Im[T]!=0||T<xL||T>xR,Abort[]]],
    Null];
    Frac=FracPer*Frac;
  ];
];

If[VectorQ[FR],T/.FR]

(*End SearchForT*)
]

(*Time from turning-circle entry point to turning circle exit point, tC*)
tC[T_,eps_,n_]:=If[
  Dot[xout[T,eps,n]-xin,up]>0,
  T-tL[T,eps,n],
  2*Pi-(T-tL[T,eps,n])
]

(*Pursuer's intercept times: T=tC+tL, correcting for the "2*Pi effect"*)
TotalTime[T_,eps_,n_]:=N[tC[T,eps,n]+tL[T,eps,n]]

(*The arcs of the pursuer's turning circles*)
PathArc[T_,eps_,n_]:=Module[{ThIn},
If[Dot[xin-xc[n],j]>0,
  ThIn=ArcCos[Dot[xin-xc[n],i]],
  ThIn=2*Pi-ArcCos[Dot[xin-xc[n],i]]
];
If[EvenQ[n],
  {ThIn,ThIn+tC[T,eps,n]},
  {ThIn-tC[T,eps,n],ThIn}
]
]

(*Define graphics objects*)

Circles[T_,eps_,n_,colour_]:=
Graphics[{Thick,colour,Circle[{xc[n][[1]],xc[n][[2]]},1,PathArc[T,eps,n]]}]

PursuerInitial:=
Graphics[{Thick,Arrowheads[Medium],
Arrow[{{xin[[1]]-up[[1]],xin[[2]]-up[[2]]},{xin[[1]],xin[[2]]}}]}]
```

```
InterceptPoint[T_,eps_,n_,colour_]:=
Graphics[{Thick,colour,Arrowheads[Medium],
Arrow[{{xout[T,eps,n][[1]],xout[T,eps,n][[2]]},{xI[T,eps][[1]],xI[T,eps][[2]]}}]}]

TargetHeading[T_,eps_]:=
Graphics[{Thick,Arrowheads[Medium],
Arrow[{{xs[[1]],xs[[2]]},{xI[T,eps][[1]],xI[T,eps][[2]]}}]}]

(*The possible path(s) and time(s) to interception: eps=cs/ca*)
TurningCircle[eps_,acinit_,shinit_,acvel_,shvel_,OptionsPattern[]]:=Module[

(*Local Variables*)
{Disp,OutputStyle,PathStyle,T0,T1},

(*Options*)
If[OptionValue[BlackAndDashed]==True,
  OutputStyle={"Solid","Dashed"};
  PathStyle={Black,Dashed};,
  (*Default*)
  OutputStyle={"Blue","Red"};
  PathStyle={Blue,Red};
];

(*Pursuer’s entry point at t=0. Global Variable.*)
xin=Append[acinit,0];

(*Target’s position at t=0. Global Variable.*)
xs=Append[shinit,0];

(*Pursuer’s velocity at t=0. Note: |up|==1. Global Variable.*)
up=Append[acvel,0];

(*Target’s velocity. Note: |us|==1. Global Variable.*)
us=Append[shvel,0];

(*Numerical Tolerance. Global Variable.*)
NumTol=10^(-6);

T0=Interception[eps,0];
T1=Interception[eps,1];

(*Generate graphics*)
Which[
!NumberQ[T0]&&NumberQ[T1],
Print[OutputStyle[[2]]," intercept time: ",TotalTime[T1,eps,1]];
Print[OutputStyle[[2]]," intercept point: ",N[{xI[T1,eps][[1]],xI[T1,eps][[2]]}]];
Print[OutputStyle[[2]]," exit point: ",N[{xout[T1,eps,1][[1]],xout[T1,eps,1][[2]]}]];
Disp=Show[InterceptPoint[T1,eps,1,PathStyle[[2]]],TargetHeading[T1,eps],
Circles[T1,eps,1,PathStyle[[2]]],PursuerInitial,
Graphics[Circle[{xc[0][[1]],xc[0][[2]]}]]];,

NumberQ[T0]&&!NumberQ[T1],
Print[OutputStyle[[1]]," intercept time: ",TotalTime[T0,eps,0]];
Print[OutputStyle[[1]]," intercept point: ",N[{xI[T0,eps][[1]],xI[T0,eps][[2]]}]];
Print[OutputStyle[[1]]," exit point: ",N[{xout[T0,eps,0][[1]],xout[T0,eps,0][[2]]}]];
Disp=Show[InterceptPoint[T0,eps,0,PathStyle[[1]]],TargetHeading[T0,eps],
Circles[T0,eps,0,PathStyle[[1]]],PursuerInitial,
Graphics[Circle[{xc[1][[1]],xc[1][[2]]}]]];,

NumberQ[T0]&&NumberQ[T1],
Print[OutputStyle[[1]]," intercept time: ",TotalTime[T0,eps,0]];
Print[OutputStyle[[2]]," intercept time: ",TotalTime[T1,eps,1]];
Print[OutputStyle[[1]]," intercept point: ",N[{xI[T0,eps][[1]],xI[T0,eps][[2]]}]];
Print[OutputStyle[[2]]," intercept point: ",N[{xI[T1,eps][[1]],xI[T1,eps][[2]]}]];
Print[OutputStyle[[1]]," exit point: ",N[{xout[T0,eps,0][[1]],xout[T0,eps,0][[2]]}]];
Print[OutputStyle[[2]]," exit point: ",N[{xout[T1,eps,1][[1]],xout[T1,eps,1][[2]]}]];
Disp=Show[InterceptPoint[T0,eps,0,PathStyle[[1]]],
InterceptPoint[T1,eps,1,PathStyle[[2]]],TargetHeading[T0,eps],
TargetHeading[T1,eps],Circles[T0,eps,0,PathStyle[[1]]],
Circles[T1,eps,1,PathStyle[[2]]],PursuerInitial];,

!NumberQ[T0]&&!NumberQ[T1],
Message[TurningCircle::err5]; Abort[];

];

(*Display graphics*)
Disp

]

End[]

EndPackage[]
```

Appendix E The points $\mathbf{x}_{\text{in}}$ and $\mathbf{x}_{\text{out}}$ in polar coordinates centred on $\mathbf{x}_c$

It may be useful to express the points $\mathbf{x}_{\text{in}}$ and $\mathbf{x}_{\text{out}}$ in polar coordinates centred on $\mathbf{x}_c$, that is, find values of θ_{in} and θ_{out} such that

$$\mathbf{x}_{\text{in}} = \mathbf{x}_c(n) + (\cos(\theta_{\text{in}}(n)), \sin(\theta_{\text{in}}(n))),$$

$$\mathbf{x}_{\text{out}}(\widetilde{T}, n) = \mathbf{x}_c(n) + (\cos(\theta_{\text{out}}(\widetilde{T}, n)), \sin(\theta_{\text{out}}(\widetilde{T}, n))),$$

where θ_{in} and θ_{out} depend on which direction the pursuer turns ($n = 0$ or $n = 1$), and $\widetilde{T}$ is the minimum time to interception returned by *Interception*; see Figure 3. Values for θ_{in} and θ_{out} can be calculated using the arctan function, however this may not take into account which quadrants of $\mathbf{x}_c(n)$ the points $\mathbf{x}_{\text{in}}$ and $\mathbf{x}_{\text{out}}$ are in. Alternatively, expressions for θ_{in} and θ_{out} can be deduced from Figure 1 such that the resulting points are in the correct quadrant. These expressions are

$$\theta_{\text{in}}(n) = \begin{cases} \arccos((\mathbf{x}_{\text{in}} - \mathbf{x}_c(n)) \cdot \mathbf{i}), & (\mathbf{x}_{\text{in}} - \mathbf{x}_c(n)) \cdot \mathbf{j} > 0 \\ 2\pi - \arccos((\mathbf{x}_{\text{in}} - \mathbf{x}_c(n)) \cdot \mathbf{i}), & (\mathbf{x}_{\text{in}} - \mathbf{x}_c(n)) \cdot \mathbf{j} \leqslant 0, \end{cases}$$

since $|\mathbf{x}_{\text{in}} - \mathbf{x}_c(n)| = 1$, where $\mathbf{i} = (1, 0, 0)$ and $\mathbf{j} = (0, 1, 0)$, and

$$\theta_{\text{out}}(\widetilde{T}, n) = \theta_{\text{in}}(n) + (-1)^n \, \widetilde{t}_c(\widetilde{T}),$$

where $\widetilde{t}_c$ is given by Equation (28).

By definition, the direction of increasing polar angle is anticlockwise. Therefore, since the pursuer moves in an anticlockwise direction on one turning-circle and in a clockwise direction on the other turning-circle, it can be shown that

$$\begin{cases} \theta_{\text{in}}(0) \leqslant \theta_{\text{out}}(\widetilde{T}, 0), & n = 0 \\ \theta_{\text{out}}(\widetilde{T}, 1) \leqslant \theta_{\text{in}}(1), & n = 1. \end{cases}$$

Page classification: UNCLASSIFIED

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. CAVEAT/PRIVACY MARKING

2. TITLE Minimum Paths to Interception of a Moving Target when Constrained by Turning Radius	3. SECURITY CLASSIFICATION Document (U) Title (U) Abstract (U)
4. AUTHOR Jason R. Looker	5. CORPORATE AUTHOR Defence Science and Technology Organisation 506 Lorimer St, Fishermans Bend, Victoria 3207, Australia

6a. DSTO NUMBER DSTO–TR–2227	6b. AR NUMBER AR 014–359	6c. TYPE OF REPORT Technical Report	7. DOCUMENT DATE December, 2008

8. FILE NUMBER 2008/1072840/1	9. TASK NUMBER DS 07/245	10. SPONSOR CAOD	11. No. OF PAGES 49	12. No. OF REFS 19

13. URL OF ELECTRONIC VERSION http://www.dsto.defence.gov.au/corporate/reports/DSTO–TR–2227.pdf	14. RELEASE AUTHORITY Chief, Air Operations Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for Public Release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SOUTH AUSTRALIA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS

No Limitations

18. DSTO RESEARCH LIBRARY THESAURUS

air operations, aircraft performance, flight dynamics, interception, maritime surveillance

19. ABSTRACT

Entities in some simulations of military operations move unrealistically from point to point and are not constrained by their turning radius. The fidelity of this representation may be insufficient for operations research studies. In this paper a pursuer intercepting a target is considered, where the pursuer and target are moving at constant speeds in two dimensions and the target has a constant velocity. The minimum feasible path to interception for a given turning radius is sought. A rigorous analysis of the model constraints produced an algorithm that can be used to systematically search the feasible region for the minimum path to interception. At the core of the algorithm is a single implicit equation for the minimum time to interception. This enables the effect of turning radius to be incorporated as a constraint into simulations of military operations, improving their fidelity. The algorithm is also straightforward to implement when compared with, for example, a traditional flight dynamics model, and has a broad range of applications in path optimisation problems, the development of computer games and robotics.

Page classification: UNCLASSIFIED